EMC Fibre Channel with QLogic Host Bus Adapters in the Linux x86 Environment

P/N 300-001-161
REV A03

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748 -9103
1-508 -435 -1000
www.emc.com

Copyright © 2001–2004 EMC® Corporation. All rights reserved.

Updated: November 2004

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

Trademark Information

EMC², EMC, Symmetrix, Celerra, CLARiiON, CLARalert, Documentum, HighRoad, Legato, Navisphere, PowerPath, ResourcePak, SnapView/IP, SRDF, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC Automated Networked Storage, EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, Access Logix, AutoAdvice, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CentraStar, CLARevent, Connectrix, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, Direct Matrix Architecture, EDM, E-Lab, Enginuity, FarPoint, FLARE, GeoSpan, InfoMover, MirrorView, NetWin, OnAlert, OpenScale, Powerlink, PowerVolume, RepliCare, SafeLine, SAN Architect, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, Universal Data Tone, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.

Contents

Preface vii

Chapter 1 Introduction

EMC/Linux Fibre Channel Environment 1-2
- Host Connectivity 1-2
- Boot Device Support 1-2
- Addressing 1-2
- Zoning 1-2
- EMC Storage Array Configuration 1-3
- Useful Linux Utilities and Functions 1-3

Chapter 2 Installing and Configuring the HBA and Driver

Installing the HBA 2-2
- Special Instructions for CLARiiON CX200 Direct-Connect Dual-Host Oracle*9i* RAC and Red Hat Enterprise Linux 2.1 Cluster Manager Clustering Configurations 2-2
- Setting the HBA FC-AL Loop ID for CLARiiON CX200 Direct-Connect Oracle*9i* RAC and Red Hat Enterprise Linux 2.1 Cluster Configurations with QLogic QLA234x-Series HBAs 2-4

Configuring the BIOS Settings 2-6
- Verifying the Correct BIOS Version 2-6
- EMC HBA Settings 2-7
- Upgrading the HBA BIOS Using the NVRAM File 2-8
- Manually Setting the QLA23xx Data Rate 2-9

Setting the Topology for QLogic HBAs 2-11
Configuring the Host with the QLogic HBA Driver 2-12
Installation Instructions for the QLogic v6.x-Series Driver 2-13

SuSE SLES 8 and QLogic v6.05.00 Driver 2-13
RHEL 2.1 and the QLogic v6.04.01 Driver 2-14
RHEL 3.0 U2 and the QLogic v6.07.02-RH2 Driver..... 2-15
Installation Instructions for the QLogic v7.x-Series Driver..... 2-17
Installing the QLogic v7.03.00 Driver 2-17
Installing the QLogic v7.03.00 Driver, Method One..... 2-18
Installing the QLogic v7.03.00 Driver, Method Two 2-22
Installing the QLogic v7.03.00 Driver, Method Three..... 2-25
Installing SNIA API Libraries with the QLogic v7.x-Series Driver 2-27
Enabling Persistent Binding for the QLogic Linux Driver 2-28
Performance Enhancements for the QLogic Linux Driver 2-30
Required Kernel v2.4.x Modifications 2-32
(CLARiiON only) Required Patch for SuSE SLES 8 SP2a. 2-32
(CLARiiON Only) Required /etc/modules.conf Edit for Red Hat Enterprise Linux 2.1 and 3.0 and SLES 8.0 Default Modular Kernels..... 2-33
(Symmetrix Only) Increasing the SCSI Generic Buffer 2-35
Additional Notes 2-37
Linux Device-Naming Convention..... 2-37
Dynamic Device Reconfiguration: Procedures 2-40
Dynamic Device Reconfiguration: Device Numbering 2-42
Linux Naviagent Limitation with fdisk on CLARiiON 2-43
Linux Naviagent Limitation with VERITAS Volume Manager on CLARiiON..... 2-46
Linux Naviagent Requirement for 64-bit Intel Systems on CLARiiON 2-47
Compaq Server-Specific Note..... 2-47
(CLARiiON Only) Disconnected Ghost LUNs..... 2-48
(CLARiiON Only) Passive-Not-Ready/ FailOverMode SP Bit Setting and Spinning Up Devices 2-49
Configuring and Compiling the Kernel 2-51
New Installation 2-51
Incorporating EMC Fibre Channel..... 2-54

Chapter 3 Configuring a Boot Device on the EMC Storage Aray

Cautions and Restrictions..... 3-2
Limitations..... 3-3
Preparing the CLARiiON Storage System 3-5
Configuring the Boot BIOS..... 3-6
Installing Red Hat Enterprise Linux 2.1 onto the Boot Device with the QLogic v6.x Series Driver 3-8

Building a Driver Disk 3-8
Installing Red Hat Enterprise Linux 2.1 onto the Storage Array 3-13
Upgrading Kernel Errata 3-14

Index i-1

Preface

Audience

This document is intended to aid in installing and configuring an EMC-qualified QLogic host bus adapter and driver in a Linux host connected to an EMC Symmetrix or CLARiiON storage array.

Conventions Used in This Guide

EMC uses the following conventions for notes and cautions.

A note presents information that is important, but not hazard-related.

CAUTION

A caution contains information essential to avoid data loss or damage to the system or equipment. The caution may apply to hardware or software.

Typographical Conventions

EMC uses the following type style conventions in this guide:

AVANT GARDE	Keystrokes
Palatino, bold	◆ Dialog box, button, icon, and menu items in text ◆ Selections you can make from the user interface, including buttons, icons, options, and field names
Palatino, italic	◆ New terms or unique word usage in text ◆ Command line arguments when used in text ◆ Book titles

Courier, italic	Arguments used in examples of command line syntax.

Courier	System prompts and displays and specific filenames or complete paths. For example:
	`working root directory [/user/emc]:`
	`c:\Program Files\EMC\Symapi\db`

Courier, bold	◆ User entry. For example: **`symmpoll -p`** ◆ Options in command line syntax

1

Introduction

This document describes the procedures for installing an EMC-approved QLogic host bus adapter (HBA) into a Linux x86 host environment and configuring the host for connection to an EMC storage array over Fibre Channel.

- EMC/Linux Fibre Channel Environment 1-2

EMC/Linux Fibre Channel Environment

Host Connectivity

This section provides an overview of supported hosts and host bus adapters. Review the *EMC Support Matrix* or contact your EMC representative for the latest information on qualified HBAs and drivers.

EMC does not currently support mixing different types of Fibre Channel HBA (including different types from the same vendor) in a server.

The Fibre Channel HBA driver functions as a device driver layer below the standard Linux SCSI adapter driver. The Fibre Channel interface is therefore transparent to the Linux disk administration system.

Boot Device Support

Linux hosts have been qualified for booting from EMC storage array devices interfaced through Fibre Channel as described in Chapter 3.

Addressing

- Linux uses SCSI-2 device access protocol in addressing its Fibre Channel devices. SCSI-3 Array Controller addressing modes are not supported. The peripheral device access mode is supported in Symmetrix® to address up to 128 logical devices, in an address range of 0 to 255.
- (EMC Symmetrix only) If you attempt to access existing devices previously interfaced via SCSI, your raw device addressing may be changed after upgrading to Fibre Channel directors.

Zoning

When using Linux hosts in a fabric environment, the zoning *must* be set up as single-initiator zoning. A single initiator zone is composed of one HBA and one or more EMC storage array ports. Storage arrays ports may be shared amongst HBAs; however, each HBA must be in its own zone.

Multi-initiator zones are not supported in a Linux fabric environment.

EMC Storage Array Configuration

EMC storage array configuration is performed by an EMC Customer Engineer (CE).

Refer to the *EMC Support Matrix* for required bit settings on Symmetrix Fibre Channel directors.

To verify that the Linux host can see all of the target EMC devices, configure the host as described Chapter 2, and use the command `more /proc/scsi/scsi`. This command will provide a listing of the attached EMC storage array devices.

Useful Linux Utilities and Functions

These utilities and functions can be helpful in performing configuration operations:

Utility/Function	Description
fdisk	Command used to create and manipulate Linux partition tables
fsck	Command used to check and repair a Linux file system
mkfs	Command used to create Linux file system on a device
mount	Command used to attach the file system on a device to the file tree
umount	Command used to detach a file system
shutdown	Command used to shut down the system gracefully
reboot	Command used to stop and restart the operating system
insmod	Utility used to dynamically load a single module into a running kernel
rmmod	Utility used to unload loadable modules from the running kernel if they are not in use and if other modules are not dependent upon those being removed
modprobe	Utility used to load or remove a set of modules that can be either a single module or a stack of dependent modules
lsmod	Utility used to list the currently loaded modules.
lspci	Utility used to display information about all of the system's PCI buses and all of the devices connected to those buses.

2

Installing and Configuring the HBA and Driver

This chapter describes the procedures for installing an EMC-approved QLogic host bus adapter (HBA) into a Linux host and configuring the host for connection to an EMC storage array over Fibre Channel.

- Installing the HBA 2-2
- Configuring the BIOS Settings 2-6
- Setting the Topology for QLogic HBAs 2-11
- Configuring the Host with the QLogic HBA Driver 2-12
- Installation Instructions for the QLogic v6.x-Series Driver 2-13
- Installation Instructions for the QLogic v7.x-Series Driver 2-17
- Enabling Persistent Binding for the QLogic Linux Driver 2-28
- Performance Enhancements for the QLogic Linux Driver 2-30
- Required Kernel v2.4.x Modifications 2-32
- Additional Notes 2-37
- Configuring and Compiling the Kernel 2-51
- Incorporating EMC Fibre Channel 2-54

Installing the HBA

Before the QLogic HBA is installed, the host must be configured with Linux. Install the Linux kernel (including sources) from the distribution installation CD, following the procedure provided in the distribution installation guide. Partition the boot drive and select the packages and services necessary for the host.

To install the QLogic HBA, follow the instructions included with your adapter. The adapter installs into a single slot.

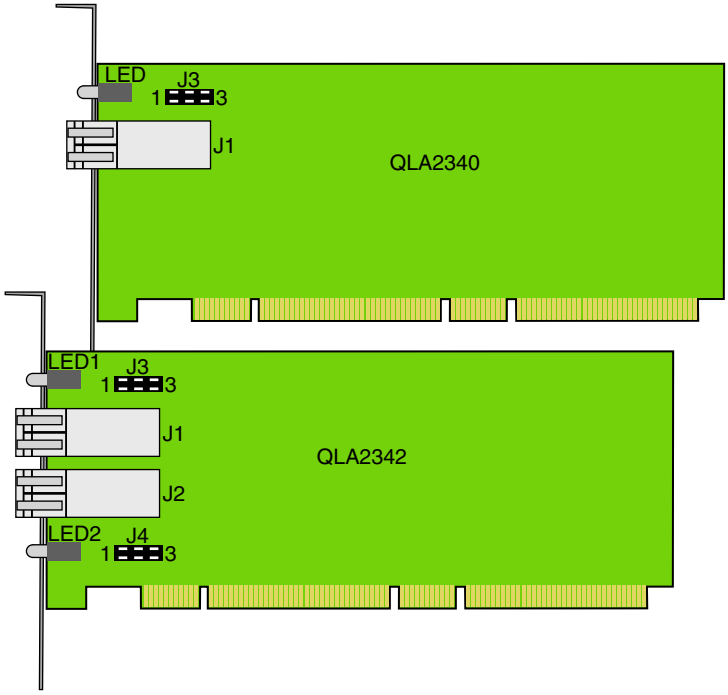
Special Instructions for CLARiiON CX200 Direct-Connect Dual-Host Oracle*9i* RAC and Red Hat Enterprise Linux 2.1 Cluster Manager Clustering Configurations

For CLARiiON® CX200 direct-connect dual-host Oracle*9i* RAC or Red Hat Enterprise Linux 2.1 Cluster Manager cluster configurations with QLA234x HBAs, you must change the default HBA optic jumper position:

CAUTION

Modifying the jumper setting without using the recommended firmware/drivers can cause the HBA to lose connectivity.

1. Remove the HBA from the server as instructed by your server guide.
2. Locate jumper(s) J3 (QLA2340) or J3 and J4 (QLA2342), shown in the following figure.



3. Move the jumper(s), onto pins 1–2 (if not already there).

If later you wish to return the jumper(s) to the default factory position, return the jumper(s) to pins 2–3.

Setting the HBA FC-AL Loop ID for CLARiiON CX200 Direct-Connect Oracle*9i* RAC and Red Hat Enterprise Linux 2.1 Cluster Configurations with QLogic QLA234x-Series HBAs

While configuring a QLA234x HBA for Red Hat 2.1 Advanced Server and connection to EMC CLARiiON CX200 for direct-connect Oracle*9i* RAC or Red Hat Enterprise Linux 2.1 cluster configurations, you must manually set the HBA FC-AL Loop ID. Follow these steps to enable loop hard addressing and set the loop ID on each HBA:

Perform this procedure on both nodes in the cluster connected to the CX200.

The topology must also be changed from the default **Point to Point** to **Loop Only**. Refer to *Setting the Topology for QLogic HBAs* on page 2-11 for instructions.

1. Boot the Red Hat host; press CTRL-Q when prompted to `Press <CTRL-Q> for Fast!UTIL`. (This prompt appears with the QLogic startup banner.)
2. After the Fast!UTIL program loads, the initial display depends on whether there are multiple QLogic HBAs installed in the server:
   - If there is only one QLogic HBA, the **Fast!UTIL Options** menu appears.
   - If there are multiple QLogic HBAs, a list of memory addresses occupied by those HBAs appears.

     Using the arrow keys, select the desired HBA and press ENTER. The **Fast!UTIL Options** menu appears.
3. Select **Configuration Settings** from the **Fast!UTIL Options** menu, and press ENTER.
4. Select **Adapter Settings** from the **Configuration Settings** menu.
5. Select **Adapter Hard Loop ID** from the Host Adapter Settings menu and press ENTER until the value is **Enabled**.
6. Select **Hard Loop ID** and press ENTER.

   The loop ID value entered here is the decimal representation of the FC-AL loop ID. You do not need to perform any FC-AL AL_PA hexadecimal translation.

HBAs connected to the same SP should all have unique hard loop IDs. For example, select **0** for all HBAs on one node, and select **1** for all HBAs on the other node.

7. Enter a value for the loop ID and press ENTER.
8. Press ESC to return to the **Configuration Settings** menu.
9. Press ESC to return to the **Fast!UTIL Options** menu.
10. When prompted to save changes made to the current adapter, select **Save Changes** and press ENTER.
11. If there are more adapters to configure, choose **Select Host Adapter**, and repeat steps 3 through 11 for each adapter.
12. Press ESC to exit Fast!UTIL.
13. Reboot the host.
14. Repeat this procedure on all nodes in the cluster connected to the CX200.

CAUTION

Future use of the firmware NVRAM file to apply settings will overwrite and possibly invalidate the changes made above. If you will later update using a firmware NVRAM file, be sure it is listed and supports this CX200 direct-connect cluster configuration.

To restore EMC default HBA settings, either reload the approved EMC NVRAM file, or return to step 5 and ensure that **Adapter Hard Loop ID** is set to **Disabled**.

Configuring the BIOS Settings

After the HBA is installed, follow these steps to verify and configure HBA firmware settings.

Refer to the *EMC Support Matrix* for required BIOS versions for qualified HBAs.

Verifying the Correct BIOS Version

You can determine the BIOS version at boot time from the QLogic banner or from the **Fast!Util Options** menu.

- From the banner — Boot the host and watch for the banner (shown in the QLA2340 example below, with the ROM BIOS version highlighted):

```
Qlogic Corporation
QLA2312 PCI Fibre Channel ROM BIOS Version 1.42
Copyright © Qlogic Corporation 1993-2002 All rights reserved
Press <CTRL - Q> for Fast!UTIL
www.qlogic.com
```

  - If the banner displays the required version, continue to *Configuring the Host with the QLogic HBA Driver* on page 2-12.
  - If the banner does not display the required version, upgrade the firmware as described under *Upgrading the HBA BIOS Using the NVRAM File* on page 2-8; then proceed to *Configuring the Host with the QLogic HBA Driver* on page 2-12.

- From the **QLogic Fast!Util Options** menu:

  a. Boot the host. As soon as the QLogic banner appears, press CTRL-Q to enter Fast!Util.

  b. Select **Configuration Settings** from the **Fast!Util Options** menu.

  c. Select **Adapter Settings** from the **Configuration Settings** menu.

  d. Under **Adapter Settings**, note the BIOS version:

  – If the banner displays the required version, continue to *Configuring the Host with the QLogic HBA Driver* on page 2-12.

  – If the banner does not display the required version, upgrade the firmware as described under *Upgrading the HBA BIOS Using the NVRAM File*; then proceed to *Configuring the Host with the QLogic HBA Driver*.

EMC HBA Settings

EMC requires configuring the QLogic BIOS settings with the EMC-approved NVRAM settings file. This file contains all of the BIOS settings for the QLogic adapters that have been tested and approved for connection to EMC storage arrays.

Refer to the *EMC Support Matrix* for required NVRAM versions for qualified HBAs.

Pre-Configured Settings

The following parameters have been pre-configured in the EMC NVRAM. They are also configurable in the **Host Adapter Settings**, **Advanced Settings**, and **Extended Firmware Settings** menus.

These settings apply to PowerPath and non-HA connections.

Table 2-1 QLA23xx BIOS Settings

Host Adapter Settings	QLogic Default	EMC Default
Host Adapter BIOS	Disabled	Disabled
Frame Size	2048	2048
Loop Reset Delay	5	5
Adapter Hard Loop ID	Disabled	Disabled
Hard Loop IP	0	0
Spinup Delay	Disabled	Disabled
Connection Options	2	2
Fibre Channel Tape Support	Enabled	Disabled
Data Rate	2 (Auto Select)	2 (Auto Select)
Advanced Adapter Settings	**QLogic Default**	**EMC Default**
Execution Throttle	16	256
LUNs per Target	8	256
Enable LIP Reset	No	No
Enable LIP Full Login	Yes	Yes
Enable Target Reset	Yes	Yes
Login Retry Count	8	8

Table 2-1 QLA23xx BIOS Settings (continued)

Port Down Retry Count	8	45
Link Down Timeout	15	45
Extended Error Logging	Disabled	Disabled
RIO Operation Mode	0	0
Interrupt Delay Timer	0	0

The QLA2200 uses the same QLogic and EMC defaults as stated in Table 2-1 for the QLA23xx series HBAs, with the following exceptions:

Table 2-2 QLA2200 BIOS Setting Exceptions to Table 2-1

Parameter	QLogic Default	EMC Default
Frame Size	1024	2048
Response Timer	0	0
Interrupt Delay Timer	0	0

There is no Data Rate parameter for the QLA2200.

Upgrading the HBA BIOS Using the NVRAM File

Follow these steps to upgrade the QLogic BIOS using the NVRAM file:

1. Obtain the latest version of the BIOS from the QLogic website:
   a. Access `http://www.qlogic.com`.
   b. Click **Downloads** at the left side of the screen.
   c. Click the **EMC** link to the right of **OEM and ISV approved/recommended drivers and firmware**.
   d. Find the NVRAM file for the HBA(s) in your system, and click the associated **Download** link to save the file.
2. The file is a self-extracting .zip file that must be extracted to a bootable diskette drive, as follows:
   a. Insert a diskette into a Microsoft Windows 9x machine.
   b. Open any DOS window.

c. At the DOS prompt, format the diskette:
`format /s a:`

d. At the DOS prompt, change (**`cd`**) to the directory in which you saved the zipped file, then extract the file to the diskette.

The QLogic HBA(s) must be installed in the host before proceeding.

3. Insert the BIOS upgrade installation diskette into the diskette drive.
4. Reboot the host.
5. After the host has rebooted, a DOS prompt appears. Type **`flasutil /L /F`** and press ENTER.

   The HBA BIOS upgrade might take a few minutes.

6. After the upgrade is complete, remove the diskette and reboot the host. During startup, the QLogic banner should display the required BIOS version listed under *Verifying the Correct BIOS Version* on page 2-6.

Manually Setting the QLA23xx Data Rate

The EMC default setting for the data rate on the QLA23xx HBAs is Auto Select mode. If necessary, the mode may be set manually to 1GB, 2GB, or Auto Select mode.

For any device connected to the HBA, set the device data rate (if applicable) before setting the HBA data rate.

For every QLA23xx HBA on which you want to manually set the HBA data rate, follow these steps:

1. Boot the host.

   When the QLogic banner appears, press CTRL-Q to enter Fast!Util.

2. Select **Configuration Settings** from the **Fast!Util Options** menu.
3. Select **Adapter Settings** from the **Configuration Settings** menu.
4. Select the **Data Rate** setting and press ENTER.
5. Select the appropriate speed for the device to which the QLA23xx HBA connects. EMC recommends using the default setting of **Option 2, Auto Select**.
6. Press ESC to return to the **Fast!Util Options** menu.

7. When prompted to save the changes made to the current adapter, select **Save Changes**.
8. Repeat steps 2 through 7 for each adapter.
9. Press ESC to ext Fast!Util.
10. Reboot the host.

Setting the Topology for QLogic HBAs

The EMC default setting for the topology currently is set to 2 (Loop preferred; otherwise, point to point). Follow these steps to set the NVRAM variables for the topology:

1. Boot the host. When the QLogic banner appears, press CTRL-Q to enter Fast!Util.
2. Select **Configuration Settings** from the **Fast!Util Options** menu.
3. Select **Adapter Settings** from the **Configuration Settings** menu.
4. Select **Connection Options** and press ENTER.
5. Select the appropriate topology and press ENTER:
   - FC-AL — **0** (loop only)
   - FC-SW — **1** (point to point only)
6. Press ESC to return to the **Fast!Util Options** menu.
7. When prompted to save the changes made to the current adapter, select **Save Changes**.
8. If you have more adapters to configure, choose **Select Host Adapter** and repeat steps 2 through 7 for each adapter.
9. Press ESC to exit Fast!Util.
10. Reboot the host.

Configuring the Host with the QLogic HBA Driver

Using the QLogic adapter with the Linux operating system requires HBA driver software. The driver functions at a layer below the Linux SCSI driver to present Fibre Channel devices to the operating system as if they were standard SCSI devices.

EMC supports neither the deprecated qlogicfc driver nor the qla2x00 driver that are included in various distributions, unless otherwise specified in the *EMC Support Matrix*.

Refer to the latest *EMC Support Matrix* for specific qualified kernel versions and distributions.

Please note that the support stated in the *EMC Support Matrix* supersedes versions listed in this document.

Installation Instructions for the QLogic v6.x-Series Driver

- SuSE SLES 8 and the QLogic v6.05.00 Driver
- RHEL 2.1 and the QLogic 6.04.01 Driver
- RHEL 3.0 and the QLogic v6.07.02 Driver

SuSE SLES 8 and QLogic v6.05.00 Driver

EMC supports the QLogic v6.05.00 driver included in SuSE SLES 8 distribution. The default driver in this distribution is the v6.06.00 driver. The following procedure should be used to enable the v6.05.00 driver.

1. In order for the driver to be loaded at boot time, the driver should be listed in the /etc/sysconfig/kernel and /etc/modules.conf files.

   - Edit the `/etc/sysconfig/kernel` file:

     `vi /etc/sysconfig/kernel`

     – For QLogic QLA2200 HBAs, add the reference to the QLogic v6.05.00 qla2200-60500 driver to the `INITRD_MODULES` line:

`INITRD_MODULES="scsi_mod sd_mod mptscsih qla2200-60500 reiserfs"`

     – For QLogic QLA2300 HBAs, add the reference to the QLogic v6.05.00 qla2300-60500 to the `INITRD_MODULES` line:

`INITRD_MODULES="scsi_mod sd_mod mptscsih qla2300-60500 reiserfs"`

     Save the changes and quit from `vi`.

   - Edit the `/etc/modules.conf` file:

     `vi /etc/modules.conf`

     – Add the entry for each installed QLogic QLA2200 HBA:

     `alias scsi_hostadapter`*`N`* `qla2200-60500`

     – Add the entry for each installed QLogic QLA2300 HBA:

     `alias scsi_hostadapter`*`N`* `qla2300-60500`

     where *N* is the sequential value of each QLogic HBA installed in the system, beginning with the number after the last host adapter number entry in the file. (The first host adapter entry begins with zero.)

Ensure that the QLogic HBA entries are listed after the internal SCSI host adapter entry in both files.

Example:

```
alias parport_lowlevel parport_pc
alias scsi_hostadapter sym53c8xx
alias scsi_hostadapter1 qla2300-60500
alias scsi_hostadapter2 qla2300-60500
alias eth0 tlan
options scsi_mod max_scsi_luns=255 scsi_allow_ghost_devices=1
```

Save the changes and quit from `vi`.

2. Create a new ramdisk to reflect the changes made:

   `cd /boot`
   `mkinitrd -k vmlinuz-`*`$1`* **`-i initrd-`*`$1`*

   where *`$1`* is the currently running v2.4.x kernel version.

   Example:

   ```
   cd /boot
   mkinitrd -k vmlinuz-2.4.21-190 -i initrd-2.4.21-190
   ```

3. Reboot the system.

RHEL 2.1 and the QLogic v6.04.01 Driver

EMC supports the QLogic v6.04.01 driver included in RHEL 2.1 distribution. In RHEL 2.1 kernels v2.4.9-e.24 and later, the QLogic v6.04.01 driver is the default driver within the kernel. The following procedure should be used to enable the v6.04.01 driver.

1. Ensure that the `/etc/modules.conf` file references an entry for each installed QLogic QLA2200 or QLA2300 series HBA.

   - For each installed QLogic QLA2200 HBA, add an entry:

     `alias scsi_hostadapterN qla2200`

   - For each installed QLogic QLA2300 HBA, add an entry:

     `alias scsi_hostadapterN qla2300`

     where `N` is the sequential value of each QLogic HBA installed in the system, beginning with the number after the last host adapter number entry in the file. (The first host adapter entry begins with zero.)

Example:

```
alias parport_lowlevel parport_pc
alias scsi_hostadapter sym53c8xx
alias scsi_hostadapter1 qla2300
alias scsi_hostadapter2 qla2300
alias eth0 tlan
options scsi_mod max_scsi_luns=255 scsi_allow_ghost_devices=1
```

2. If any modifications are made to the `/etc/modules.conf` file to include references to the QLogic HBAs, create a new ramdisk to reflect the changes made:

   `cd /boot`
   `mkinitrd –v initrd-`*`$1`*`.img` *`$1`*

   where *`$1`* is the currently running v2.4.x kernel version.

   Example:

   ```
   cd /boot
   mkinitrd –v initrd-2.4.9-e.38smp.img 2.4.9-e.38smp
   ```

3. Reboot the host.

RHEL 3.0 U2 and the QLogic v6.07.02-RH2 Driver

EMC supports the QLogic v6.07.02-RH2 driver included in RHEL 3.0 U2 distribution. In RHEL 3.0 kernel v2.4.21-15.EL and later, the QLogic v6.07.02-RH2 driver is the default driver within the kernel. The following procedure should be used to enable the v6.07.02-RH2 driver.

1. Ensure that the `/etc/modules.conf` file references an entry for each installed QLogic QLA2200 or QLA2300 series HBA.

   - For each installed QLogic QLA2200 HBA, add an entry:

     `alias scsi_hostadapter`*`N`* **`qla2200`**

   - For each installed QLogic QLA2300 HBA, add an entry:

     `alias scsi_hostadapter`*`N`* **`qla2300`**

   where *`N`* is the sequential value of each QLogic HBA installed in the system, beginning with the number adter the last host adapter number entry in the file. (The first host adapter entry begins with zero.)

   Example:

```
alias parport_lowlevel partport_pc
alias scsi_hostadapter sym53c8xx
alias scsi_hostadapter1 qla2300
alias scsi_hostadapter2 qla2300
alias eth0 tg3
options scsi_mod max_scsi_luns=255 scsi_allow_ghost_devices=1
```

2. If any modifications are made to the `/etc/modules.conf` file to include references to the QLogic HBAs, create a new ramdisk to reflect the changes made:

   `cd /boot`
   `mkinitrd -v initrd-`*`$1`***`.img`** *`$1`*

   where *`$1`* is the currently running v2.4.x kernel version.

   Example:

   ```
   cd /boot
   mkinitrd –v initrd-2.4.21-15.ELsmp.img 2.4.21-15.ELsmp
   ```

3. Reboot the host.

Installation Instructions for the QLogic v7.x-Series Driver

Installing the QLogic v7.03.00 Driver

Installation Methods for the QLogic v7.03.00 Driver

The QLogic v7.03.00 driver can be installed onto a Linux host using either of two methods:

- Method One – Build the driver in the directory created for it.
- Method Two – Use the Dell-generated DKMS RPM to compile and install the modular driver for Dell servers running RHEL 2.1 and 3.0 attached to CLARiiON storage arrays.
- Method Three - Use the QLogic RPM to compile and install the modular driver for RHEL 2.1, RHEL 3.0, and SLES 8 systems attached to AX100 arrays.

Here is a brief description of each method of installation for the v7.03.00 driver:

- Method One – Manually build the driver in the directory created for it.
- Method Two – Requires no manual edits for Dell servers running RHEL 2.1 and 3.0 attached to CLARiiON CXx00 series storage arrays. By installing the Dell-generated DKMS RPM, the necessary files will be edited and the driver will be compiled and installed automatically.
- Method Three – Requires no manual edits for systems attached to AX100 arrays. By installing the QLogic RPM, the necessary files will be edited and the driver will be compiled and installed automatically.

Downloading the Driver

Use the following procedure to download the QLogic driver from the QLogic website:

1. Create a directory to which you will save the driver:

   ```
   mkdir /home/qlogic
   ```

2. Use a web browser to access http://www.qlogic.com.
3. Click **Downloads** at the left side of the screen.

4. Click the **EMC** link to the right of **OEM and ISV approved/recommended drivers and firmware**.
5. Find the supported driver for your kernel version, and click the associated **Download** link to save the file to the local directory you created.

What Next? Select a method to install the driver onto the Linux host:

- For a manually installed v7.03.00 driver, refer to *Installing the QLogic v7.03.00 Driver, Method One* on page 2-18.
- For a modular v7.03.00 driver on a Dell server attached to a CLARiiON CXx00 series storage array, refer to *Installing the QLogic v7.03.00 Driver, Method Two* on page 2-22.
- For a modular v7.03.00 driver on a server attached to a CLARiiON AX100 storage array, refer to *Installing the QLogic v7.03.00 Driver, Method Three* on page 2-25.

Installing the QLogic v7.03.00 Driver, Method One

Follow these steps to configure the driver as modular only:

1. Change to the directory to which the tar archive was saved:

   ```
   cd /home/qlogic
   ```

2. Uncompress and extract the source files from the tar archive:

   ```
   tar zxvf qla2x00-v7.03.00-dist.tgz
   ```

   The initial uncompression will provide you with the following:

   ```
   qlogic/
   qlogic/drvrinstall
   qlogic/libinstall
   qlogic/libremove
   qlogic/qlapi-v2.01beta5-rel.tgz
   qlogic/qla2x00src-v7.03.00.tgz
   readme.txt
   ```

3. Uncompress and extract the driver sources from the tar archive:

   ```
   cd qlogic/
   tar zxvf qla2x00src-v7.03.00.tgz
   ```

   The files are:

```
BUILD_KERNEL.txt
Config.in
exioct.h
exioctln.h
inioct.h
listops.h
makefile
Makefile.kernel
ql2100_fw.h
ql2200_fw.h
ql2200ip_fw.h
ql2300flx_fw.h
ql2300ipx_fw.h
ql2322flx_fw.h
ql2322ipx_fw.h
qla2100.c
qla2200.c
qla2200_conf.c
qla2300.c
qla2300_conf.c
qla2x00.c
qla2x00.h
qla2x00_ioctl.c
qla_cfg.c
qla_cfg.h
qla_cfgln.c
qla_debug.h
qla_devtbl.h
qla_fo.c
qla_fo.cfg
qla_fo.h
qla_gbl.h
qla_gs.c
qla_gs.h
qla_inioct.c
qla_inline.h
qla_ip.c
qla_ip.h
qla_mbx.c
qla_mbx.h
qla_opts.c
qla_opts.h
qla_ppc64.c
qla_settings.h
qla_vendor.c
qla_version.h
qlfo.h
qlfolimits.h
qlfoln.h
README.qla2x00
release.txt
revision.notes
```

SUPPORTED_KERNEL_VERSION.txt

4. To build the driver, enter one of the following commands depending upon your distribution and system type:

   - The link `/usr/src/linux-2.4` exists by default on Red Hat distributions so the following appropriate command may be used:

For Red Hat Distributions	
System	**Enter**
UP	**`make all install`**
SMP	**`make SMP=1 all install`**

   If you are using RHEL 2.1, you must prepare your kernel for a modular driver build prior to compiling the driver.

   Note that this step does not compile the kernel, but rather it is used to prepare the kernel symbols.

   Use the following steps prior to the driver compilation:

```
cd /usr/src/linux-$1
make oldconfig && make dep
```

   where *$1* is the kernel version.

   - When using SuSE, the system does not contain a link from /usr/src/linux-$1 to /usr/src/linux-2.4, so the OSVER parameter must be used.

     If desired, you can create the link on the SuSE system so the OSVER parameter will not have to be used. To create a link, enter the following:

```
cd /usr/src
ln –s linux-$ linux-2.4
```

     where *$1* is the kernel version.

     Prior to compiling the driver, you must prepare your kernel for a modular driver build.

     Note that this step does not compile the kernel, but rather it is used to prepare the kernel symbols.

     Use the following steps prior to the driver compilation:

```
cd /usr/src/linux
```

or if you created the symbolic link, use:

```
cd /usr/src/linux-$1
make cloneconfig && make dep
```

Compile the driver:

For SuSE Distributions	
System	**Enter:**
UP	`make OSVER=linux-$1 all install`
SMP	`make SMP=1 OSVER=linux-$1 all install`

where $1 is the kernel version.

This method creates and installs both the qla2200.o and qla2300.o object files.

5. Run the following command to update the `/lib/modules/$1/modules.dep` file:

   ```
   depmod -ae
   ```

   where $1 is the currently running v2.4.x kernel version.

6. To load the QLogic driver onto the system, enter this command:

   - For the QLA2200 HBA:

     ```
     modprobe qla2200
     ```

   - For the QLA23xx HBAs:

     ```
     modprobe qla2300
     ```

7. In order to load the driver automatically at boot time, the driver should be listed in `/etc/modules.conf`:

   ```
   vi /etc/modules.conf
   ```

   - Add this entry for each installed QLogic QLA2200 HBA:

     ```
     alias scsi_hostadapterN qla2200
     ```

   - Add this entry for each installed QLogic QLA23xx HBA:

     ```
     alias scsi_hostadapterN qla2300
     ```

where *N* is the sequential value of each QLogic HBA installed in the system, beginning with the number after the last host adapter number entry in the file. (The first host adapter number entry begins with zero.)

Example:

```
alias parport_lowlevel parport_pc
alias scsi_hostadapter sym53c8xx
alias scsi_hostadapter1 qla2300
alias scsi_hostadapter2 qla2300
alias eth0 tg3
alias eth1 tg3
options scsi_mod max_scsi_luns=255 scsi_allow_ghost_devices=1
```

8. Create a new ramdisk to reflect the changes made:
   - For Red Hat distributions use:

     cd /boot
     mkinitrd –v initrd-*$1***.img** *$1*

     where *$1* is the currently running v2.4.x kernel version.

     Example:

```
cd /boot
mkinitrd –v initrd-2.4.21-15.ELsmp.img 2.4.21-15.ELsmp
```

   - For SuSE distributions use:

     cd /boot
     mkinitrd –k vmlinuz-*$1* **–i initrd-***$1*

     Example:

```
cd /boot
mkinitrd –k vmlinuz-2.4.21-190 –i initrd-2.4.21-190
```

9. Reboot the host.

Installing the QLogic v7.03.00 Driver, Method Two

This section guides you through the process of installing and utilizing the Dell DKMS RPM for the RHEL 2.1 and 3.0 distributions. The DKMS RPM will build and install the qla2200 and qla2300 drivers and modify your `/etc/modules.conf` file. The hostadapter line will be appended to the `/etc/modules.conf` file for the qla2200 and qla2300 drivers. The options line containing the addition of the **scsi_allow_ghost_devices** and **max_scsi_luns** parameters will also be appended to the file. This will allow the host to correctly identify

the disconnected LUN 0 that is reported when attached to a CLARiiON storage array as well as allow the SCSI stack to scan up to 255 devices. The Navisphere Host Agent requires that the disconnected LUN 0 be reported.

The DKMS RPM will create the QLogic driver as a module.

Follow these steps to integrate the QLogic driver on the RHEL 2.1 or 3.0 hosts:

1. Boot into the qualified and supported kernel onto which the driver will be installed.
2. Create a directory to which you will save the DKMS RPM:

   mkdir /home/qlogic

3. Obtain the `qla2x00-v7.03.00-1dkms.tgz` from the QLogic website as instructed under *Downloading the Driver* on page 2-17.

Install the DKMS RPM onto your system with the following steps:

1. Uncompress and extract the source files from the tar archive:

   tar zxvf qla2x00-v7.03.00-1dkms.tgz

   The initial decompression will provide you with the following:

   ```
   ./qla2x00-v7.03.00/
   ./qla2x00-v7.03.00/README.dkms
   ./qla2x00-v7.03.00/dkms-1.05-1.noarch.rpm
   ./qla2x00-v7.03.00/qla2x00-v7.03.00-1dkms.noarch.rpm
   ```

2. Install the DKMS RPM:

   ```
   rpm -ivh dkms-1.05-1.noarch.rpm
   Preparing...
      ########################################### [100%]
      1:dkms
      ########################################### [100%]
   ```

3. Install the QLogic driver RPM:

   rpm -ivh qla2x00-v7.03.00-1dkms.noarch.rpm

   The output from the installation of the RPM is as follows:

```
rpm -ivh qla2x00-v7.03.00-1dkms.noarch.rpm
Preparing...                ########################################### [100%]
   1:qla2x00                ########################################### [100%]

Creating symlink /var/dkms/qla2x00/v7.03.00/source ->
                 /usr/src/qla2x00-v7.03.00
```

```
DKMS: Add Completed.

Preparing kernel 2.4.21-15.ELsmp for module build:
(This is not compiling a kernel, only just preparing kernel symbols)
Running Red Hat preparation routine
make clean.....
using /lib/modules/2.4.21-15.ELsmp/build/configs/kernel-2.4.21-i686-smp.confi
g
make oldconfig.....
running dkms_mkkerneldoth....

Building module:
cleaning build area....
make KERNELRELEASE=2.4.21-15.ELsmp SMP=1 all
   INCLUDEDIR=/lib/modules/2.4.21-15.ELsmp/build/include.............
cleaning build area....

DKMS: Build Completed.

Found an original module qla2200.o in /lib/modules/
Archiving it within the DKMS tree for uninstallation purposes.

Copying the new qla2200.o for qla2x00 to
   /lib/modules/2.4.21-15.ELsmp/kernel/drivers/addon/qla2200

Found an original module qla2300.o in /lib/modules/
Archiving it within the DKMS tree for uninstallation purposes.

Copying the new qla2300.o for qla2x00 to
   /lib/modules/2.4.21-15.ELsmp/kernel/drivers/addon/qla2200
Adding static entries for this module into /etc/modules.conf
depmod....

Saving old initrd as /boot/initrd-2.4.21-15.ELsmp_old.img
Making new initrd as /boot/initrd-2.4.21-15.ELsmp.img
mkinitrd....
If your next boot fails, edit your bootloader file and revert
to the _old initrd image.

DKMS: Install Completed.
```

An example of the modified `/etc/modules.conf` file is as follows:

```
alias eth0 tg3
alias eth1 tg3
alias scsi_hostadpater aacraid
alias usb-controller usb-ohci
alias scsi_hostadapter1 qla2300
alias scsi_hostadapter2 qla2300
```

```
options scsi_mod scsi_allow_ghost_devices=1
  max_scsi_luns=255
```

A new ramdisk has automatically been created by the DKMS RPM installation.

4. Reboot the host.

Installing the QLogic v7.03.00 Driver, Method Three

This section guides you through the process of installing and utilizing the QLogic RPM. The QLogic RPM will build and install the qla2200 and qla2300 drivers and modify your `/etc/modules.conf` file. The `hostadapter` line will be appended to the `/etc/modules.conf` file for the qla2200 and qla2300 drivers and the `options` line containing the addition of the **scsi_allow_ghost_devices** and **max_scsi_luns** parameters will also be appended to the file. This will allow the host to correctly identify the disconnected LUN 0 that is reported when attached to a CLARiiON storage array as well as allow the SCSI stack to scan up to 255 devices. The Navisphere Host Agent requires that the disconnected LUN 0 be reported.

This RPM will create the QLogic driver as a module.

Follow these steps to integrate the QLogic driver on the host:

1. Boot into the qualified and supported kernel onto which the driver will be installed.
2. Create a directory to which you will save the RPM:

   `mkdir /home/qlogic`

3. Obtain the `qla2x00-v7.03.00-1.i686.rpm` from the QLogic website as instructed under *Downloading the Driver* on page 2-17.

Install the QLogic RPM onto your system with the following steps:

1. Uncompress and extract the source files from the tar archive:

```
rpm -ivh qla2x00-v7.03.00-1.i686.rpm
Preparing...                ########################################### [100%]
   1:qla2x00                ########################################### [100%]
```

The output from the installation of the RPM is as follows:

```
Qlogic qla2300 HBA Driver Installation. Supports the following
distribution and its different flavours :
Red Hat Advanced Server 2.1 / Red Hat Enterprise Linux 3.0 and SuSE  SLES 8
```

```
Installing qla2300 driver for 2.4.21-15.ELsmp...
Creating initial /usr/src/qla2x00/install.v7.03.00-1.log...
Please wait: Preparing qla2300 modular driver build

  copying kernel-2.4.21-i686-smp.config to
   /lib/modules/2.4.21-15.ELsmp/build/.config...
Building Kernel Dependencies



Make successful. File /usr/src/qla2x00/make.2.4.21-15.ELsmp.v7.03.00-1.log
   created.

install -d -o root -g root /lib/modules/2.4.21-15.ELsmp/kernel/drivers/scsi/
install -o root -g root qla2200.o qla2300.o
   /lib/modules/2.4.21-15.ELsmp/kernel/drivers/scsi/
install -o root -g root qla2200_conf.o
   /lib/modules/2.4.21-15.ELsmp/kernel/drivers/scsi/
install -o root -g root qla2300_conf.o
   /lib/modules/2.4.21-15.ELsmp/kernel/drivers/scsi/

depmod -a
make: Nothing to be done for `/lib/modules/2.4.21-15.ELsmp/kernel/drivers/scsi/'.

depmod...

mkinitrd...

Installation of qla2300 driver for 2.4.21-15.ELsmp complete.

 To load the qla2300 driver execute the following command:
 # modprobe -v qla2300
 Or Reboot the system for newly build ramdisk to load
 qla2300 driver automatically for you
_root@l82bi129:/extra [root@l82bi129 extra]# exit

Script done on Tue 04 May 2004 08:50:25 AM EDT
```

An example of the modified `/etc/modules.conf` file is as follows:

```
alias eth0 tg3
alias eth1 tg3
alias scsi_hostadpater aacraid
alias usb-controller usb-ohci
alias scsi_hostadapter1 qla2300
alias scsi_hostadapter2 qla2300
post-remove qla2200 rmmod qla2200_conf
post-remove qla2300 rmmod qla2300_conf
```

A new ramdisk is created automatically by the RPM installation.

2. Reboot the host.

Installing SNIA API Libraries with the QLogic v7.x-Series Driver

The `qla2x00-v7.03.00-dist.tgz` tar archive includes not only the driver sources, but also a kit with the necessary SNIA API libraries. If required, the SNIA API libraries may be installed by following the steps below.

1. Uncompress and extract the source files from the tar archive:

   `tar zxvf qla2x00-v7.03.00-dist.tgz`

   The initial decompression will provide you with the following:

   ```
   qlogic/
   qlogic/drvrinstall
   qlogic/libinstall
   qlogic/libremove
   qlogic/qlapi-v2.01beta5-rel.tgz
   qlogic/qla2x00src-v7.03.00.tgz
   readme.txt
   ```

2. Change to the newly created directory and uncompress the SNIA API libraries from the tar archive:

   `cd qlogic/`
   `tar zxvf qlapi-v2.01beta5-rel.tgz`

   Uncompressing the API tar archive will provide you with the following:

   ```
   doc/
   doc/readme
   hba.conf
   lib/
   lib/libqlsdm-ia32.so
   lib/libqlsdm-ia64.so
   ```

   To install the SNIA API libraries, run the installation script:

```
./libinstall
Setting up QLogic HBA API library....
Please make sure the /usr/lib/libqlsdm.so file is not in use.
Done.
```

3. The libraries are now ready to be used in conjunction with the QLogic driver.

Enabling Persistent Binding for the QLogic Linux Driver

This section provides the instructions for enabling persistent binding for the QLogic Linux driver.

For a fabric configuration, EMC highly recommends enabling persistent binding. Without a persistent binding mechanism, the host will be unable to maintain a persistent logical routing of communication from a storage devices object across the fabric to an EMC storage array volume. If the physical configuration of the switch is changed (for example, cables are swapped or the host is rebooted), the logical route becomes inconsistent, which could cause possible data corruption.

To enable persistent binding for the QLogic Linux driver, you must edit `/etc/modules.conf`. The initiator and target information necessary for persistent binding are logged in `/proc/scsi/qla2x00/n` (where *n* indicates the file for each adapter in the system) when the driver is loaded.

grep the file to obtain the initiator and target information.

For example, for a host with QLA23xx HBAs:

```
grep scsi-qla /proc/scsi/qla2300/0
```

produces output similar to the following for the first QLogic QLA23xx HBA in the system:

```
scsi-qla0-adapter-node=200000e08b036ff0;
scsi-qla0-adapter-port=210000e08b036ff0;
scsi-qla0-target-0=50060482c33ef40f;
scsi-qla0-target-1=50060482c33ef41f;
```

and

```
grep scsi-qla /proc/scsi/qla2300/1
```

produces output similar to the following for the second QLogic QLA23xx HBA in the system:

```
scsi-qla1-adapter-node=200000e08b01ae84;
scsi-qla1-adapter-port=210000e08b01ae84;
scsi-qla1-target-0=50060482c33ef40f;
scsi-qla1-target-1=50060482c33ef41f;
```

If you have QLogic QLA2200 HBAs in your system, you would specify:

```
grep scsi-qla /proc/scsi/qla2200/n
```

(where *n* indicates the file for each adapter in the system).

The information obtained from greping the appropriate files is used to assign the persistent binding information with the **ql2xopts** parameter, and is appended to `/etc/modules.conf`.

The following is an example of `/etc/modules.conf` for a v2.4.x system using the QLogic Linux driver:

The last six lines of the example are actually all one line; however, the length of the line forces it to wrap to multiple lines in the display.

```
alias parport_lowlevel parport_pc
alias eth0 eepro100
alias scsi_hostadapter cpqarray
alias scsi_hostadapter1 cpqarray
alias scsi_hostadapter2 qla2300
alias scsi_hostadapter3 qla2300
options scsi_mod max_scsi_luns=256 scsi_allow_ghost_devices=1
options qla2300 ql2xopts=scsi-qla0-adapter-port=210000e08b036ff0;scsi-qla0-tgt-
000-di-00-node=50060482c33ef40f;scsi-qla0-tgt-000-di-00-port=50060482c33ef40f;sc
si-qla0-tgt-001-di-00-node=50060482c33ef41f;scsi-qla0-tgt-001-di-00-port=5006048
2c33ef41f;scsi-qla1-adapter-port=210000e08b01ae84;scsi-qla1-tgt-000-di-00-node=
50060482c33ef40f;scsi-qla1-tgt-000-di-00-port=50060482c33ef40f;scsi-qla1-tgt-001
-di-00-node=50060482c33ef41f;scsi-qla1-tgt-001-di-00-port=50060482c33ef41f;
```

Performance Enhancements for the QLogic Linux Driver

In addition to using a more powerful server and a faster QLogic HBA, such as the QLA2340 or QLA2342, there are parameters that can be modified within the QLogic driver itself to improve performance. The values to which these parameters are set depends upon the environment and the type of applications and subsequent I/O being used on the server.

Please note that these are purely guidelines as each environment differs.

1. The queue depth parameter is the maximum number of outstanding I/Os that a device can handle. This parameter may be increased at runtime when you load the driver.

   The value of the device queue depth may be increased at driver loading time from the default of 16 to the maximum of 64.

   a. To load the QLogic driver onto the system with this parameter increased:

      – For QLA23xx HBAs, use:

      modprobe qla2300.o ql2xmaxqdepth=64

      – For QLA2200 HBAs, use:

      modprobe qla2200.o ql2xmaxqdepth=64

   b. In order for the modular driver to be loaded at boot time with the queue depth parameter changed to 64, a separate options line needs to be added to `/etc/modules.conf`.

      Edit the file:

      vi /etc/modules.conf

   Add the options line for the appropriate driver setting the queue depth to the desired setting.

      options qla2300 ql2xmaxqdepth=64

      An example of a modified /etc/modules.conf is as follows:

```
alias parport_lowlevel parport_pc
alias eth0 eepro100
alias scsi_hostadapter cpqarray
alias scsi_hostadapter1 cpqarray
alias scsi_hostadapter2 qla2300
alias scsi_hostadapter3 qla2300
options scsi_mod max_scsi_luns=255 scsi_allow_ghost_devices=1
```

```
options qla2300 ql2xmaxqdepth=64
options qla2300 ql2xopts=scsi-qla0-adapter-port=210000e08b036ff0;scsi-qla0-tgt-
000-di-00-node=50060482c33ef40f;scsi-qla0-tgt-000-di-00-port=50060482c33ef40f;sc
si-qla0-tgt-001-di-00-node=50060482c33ef41f;scsi-qla0-tgt-001-di-00-port=5006048
2c33ef41f;scsi-qla1-adapter-port=210000e08b01ae84;scsi-qla1-tgt-000-di-00-node=
50060482c33ef40f;scsi-qla1-tgt-000-di-00-port=50060482c33ef40f;scsi-qla1-tgt-001
-di-00-node=50060482c33ef41f;scsi-qla1-tgt-001-di-00-port=50060482c33ef41f;
```

After the modification to `/etc/modules.conf` has been made, a new ramdisk needs to be created and the host rebooted.

To create a new ramdisk, run the **`mkinitrd`** command:

- For Red Hat distributions use:

  ```
  cd /boot
  mkinitrd -v initrd-$1.img $1
  ```

  where *$1* is the currently running v2.4.x kernel version.

- For SuSE distributions use:

  ```
  cd /boot
  mkinitrd -k vmlinuz-$1 -i initrd-$1
  ```

  where *$1* is the currently running v2.4.x kernel version.

2. The execution throttle in the NVRAM should be set to the maximum of 256.

   - If you are using the EMC-approved QLogic BIOS/NVRAM, then the execution throttle will be defaulted to 256.
   - If you are using the QLogic default BIOS/NVRAM, then the default for this parameter is 16. In this case, you will need to:
     - Reboot the host.
     - Press <CTRL> Q when prompted.
     - Manually increase the execution throttle parameter from 16 to 256 in the NVRAM settings for each HBA in the system.

Required Kernel v2.4.x Modifications

This section describes changes you must make to Linux hosts running v.2.4.x kernels and attached to EMC storage arrays.

Refer to the appropriate sections:

- *(CLARiiON only) Required Patch for SuSE SLES 8 SP2a* on page 2-32
- *(CLARiiON Only) Required /etc/modules.conf Edit for Red Hat Enterprise Linux 2.1 and 3.0 and SLES 8.0 Default Modular Kernels* on page 2-33
- *(Symmetrix Only) Increasing the SCSI Generic Buffer* on page 2-35

(CLARiiON only) Required Patch for SuSE SLES 8 SP2a

The `rev1_sles8sp2a.patch` is required for SuSE SLES 8 (UnitedLinux) updated with the Service Pack 2a (v2.4.19-SuSE.304) when attached to CLARiiON storage arrays.

The `rev1_sles8sp2a.patch` for SuSE SLES 8 (UnitedLinux) SP2a fixes a problem in the SCSI scanning code pertaining to peripheral qualifiers, eliminates the 100-second spin-up time of a Not Ready CLARiiON device, and allows the correct identification of the disconnected LUN 0 that is reported when attached to CLARiiON storage arrays.

If the patch is not applied, then the operating system will not properly scan and recognize the attached CLARiiON devices. The contents of this patch have been incorporated by SuSE with SP3. If possible, EMC recommends upgrading from SP2a to SP3 in order to avoid applying the patch and compiling the kernel.

For example,

```
more /etc/UnitedLinux-release
UnitedLinux 1.0 (i586)
VERSION = 1.0

more /etc/SuSE-release
SuSE SLES-8 (i386)
VERSION = 8.1

uname -r
2.4.19-4GB-SMP

rpm -qf /boot/vmlinuz
```

```
k_smp-2.4.19-304
```

or

```
rpm -q kernel-source
kernel-source-2.4.19.SuSE-304
```

The files `/etc/SuSE-release` and `/etc/UnitedLinux-release`, as well as `uname -r` do show what OS is installed, but do not indicate the kernel errata that is installed. You will need to use the `rpm -q` command in order to determine the kernel errata you are running.

Applying a Patch

To apply the patch:

1. Create a directory in which the patch will reside:

   ```
   mkdir /home/patches
   ```

2. To download either patch from the EMC FTP site, use a web browser to access `ftp://ftp.emc.com/pub/elab/linux` or use FTP to directly access `ftp.emc.com`.
3. Log in to the site using a user name of **`anonymous`** and your e-mail address as the password.
4. Save the appropriate patch to the local directory you created.
5. Change to the directory from which the patch will be applied:

   ```
   cd /usr/src/linux-2.4.19.SuSE
   ```

6. Apply the patch for SLES 8 SP2a:

   ```
   patch -p1 < /home/patches/rev1_sles8sp2a.patch
   ```

What Next?

Proceed to *Configuring and Compiling the Kernel* on page 2-51.

(CLARiiON Only) Required /etc/modules.conf Edit for Red Hat Enterprise Linux 2.1 and 3.0 and SLES 8.0 Default Modular Kernels

Red Hat and SuSE have added a parameter to the SCSI stack (`scsi_scan.c` file) to allow for devices marked as *offline* to be accessible. The **`scsi_allow_ghost_devices`** parameter eliminates the need to apply the `linux-2.4.x-scsi_scan.patch` to hosts attached to the CLARiiON storage array running any of these distribution releases.

For RHEL, the **max_scsi_luns** parameter must also be enabled in

/etc/modules.conf. In the default Red Hat kernels, **CONFIG_SCSI_MULTI_LUN** is disabled. As a result, if the target 0 LUN 0 disappears from the storage group and the host is rebooted or the driver unloaded and reloaded, the host will see only the ghost LUNs (with **ArrayCommPath** disabled) or the LUNZ (with **Array-CommPath** enabled).

By default, the SuSE SLES 8 kernel has enabled the **CONFIG_SCSI_MULTI_LUN** parameter. Therefore, there is no need to add the **max_scsi_luns** parameters to the /etc/modules.conf file on SLES 8.

By appending the **scsi_allow_ghost_devices** parameter to /etc/modules.conf, a kernel recompilation will not be required in order to properly connect the Linux host to a CLARiiON storage array.

Follow these steps to enable the new parameter in /etc/modules.conf:

1. Change to the appropriate directory and edit the file:

```
cd /etc
vi /etc/modules.conf
```

2. Append the following line to the file:

   - For RHEL use:

```
options scsi_mod max_scsi_luns=255 scsi_allow_ghost_devices=1
```

Example:

```
alias parport_lowlevel parport_pc
alias scsi_hostadapter sym53c8xx
alias scsi_hostadapter1 qla2300
alias eth0 tlan
alias scsi_hostadapter2 qla2300
options scsi_mod max_scsi_luns=255 scsi_allow_ghost_devices=1
```

   - For SLES 8 use:

```
options scsi_mod scsi_allow_ghost_devices=1
```

3. Create a new ramdisk:

   - For RHEL use:

```
mkinitrd initrd-$1.img $1
```

   - For SLES 8 use:

```
mkinitrd -k vmlinuz-$1 -i initrd-$1
```

(where *$1* is the kernel version)

4. Update the appropriate bootloader configuration file prior to rebooting.

   If LILO is being used, ensure that LILO is run prior to rebooting the host in order to update the loading map:

   lilo -v

   GRUB does not have to be run even when changes are made to `/boot/grub/grub.conf`.

5. Reboot the host.

(Symmetrix Only) Increasing the SCSI Generic Buffer

For Linux hosts attached to Symmetrix storage arrays and using the SYMAPI v5.0.1 and earlier, the value for the SCSI generic buffer must be increased. If using the SYMAPI v5.02 or later, there is no need to increase this buffer or compile the kernel.

EMC highly recommends upgrading to a newer version of the SymAPI to avoid compiling the kernel.

The file `sg.h` is located in two directories, and should be modified in both. Change to the appropriate directory and increase the value for **SG_SCATTER_SZ** to the maximum of **131072**.

1. **cd /usr/src/linux/include/scsi**
2. **vi sg.h**
3. Around line 195, look for the following code block:

```
#define SG_SCATTER_SZ (8 * 4096) /* PAGE_SIZE not available to user */
/* Largest size (in bytes) a single scatter-gather list element can have.
   The value must be a power of 2 and <= (PAGE_SIZE * 32) [131072 bytes on
   i386]. The minimum value is PAGE_SIZE. If scatter-gather not supported
   by adapter then this value is the largest data block that can be
   read/written by a single scsi command. The user can find the value of
   PAGE_SIZE by calling getpagesize() defined in unistd.h . */
```

   Change the value for **SG_SCATTER_SZ (8* 4096)** to (**32 * 4096)**.

4. **cd /usr/include/scsi**
5. **vi sg.h**

6. Search for the same code block as shown in step 3. Change the value for **SG_SCATTER_SZ (8* 4096)** to (**32 * 4096)**.

Ensure that the SCSI generic driver is enabled as static when configuring the kernel on a Linux host attached to a Symmetrix storage array.

After the kernel compilation and system reboot, using the following command can validate this change:

```
cat /proc/sys/kernel/sg-big-buff
```

What Next?

Proceed to *Configuring and Compiling the Kernel* on page 2-51.

Additional Notes

This section contains the following:

- *Linux Device-Naming Convention* on page 2-37
- *Dynamic Device Reconfiguration: Procedures* on page 2-40
- *Dynamic Device Reconfiguration: Device Numbering* on page 2-42
- *Linux Naviagent Limitation with fdisk on CLARiiON* on page 2-43
- *Linux Naviagent Limitation with VERITAS Volume Manager on CLARiiON* on page 2-46
- *Linux Naviagent Requirement for 64-bit Intel Systems on CLARiiON* on page 2-47
- *Compaq Server-Specific Note* on page 2-47
- *(CLARiiON Only) Disconnected Ghost LUNs* on page 2-48
- *(CLARiiON Only) Passive-Not-Ready/ FailOverMode SP Bit Setting and Spinning Up Devices* on page 2-49

Linux Device-Naming Convention

The four high-level device drivers in Linux are:

`sd` — Direct access (disks)

`sg` — SCSI generic interface

`sr` — Data CD-ROMs

`st` — Tapes

The `sg` driver is a character-based device while the other three drivers are block-based devices. The `sg` driver is used primarily for scanners, CD writers, and printers. The `sg` device files are dynamically mapped to SCSI IDs/LUNs on the SCSI bus starting with the first SCSI controller.

Block Device File Names

A native device file name for block devices take the following form:

```
/dev/sdln
```

where:

- `l` is a letter denoting the physical drive.

- `n` is a number denoting the partition on that physical drive.

  Usually, the partition number is not included when referring to the entire drive.

Following this format, the filenames are as follows:

```
/dev/sd[a-d][a-z][1-15]
```

Linux kernel reserves 16 major numbers for SCSI devices. Each major number can have zero to 255 minor numbers. These minor numbers include the partitions for a SCSI device. Linux supports one to 15 partitions per disk device. Partitions 1 through 4 are the primary partitions, while partitions 5 and greater are the logical or extended partitions. These limitations are specific to the Intel platform. By default, slices are not used in Linux.

Therefore, the product of 16 major numbers and 16 minor numbers yields 256 SCSI devices. As a result, the kernel is able to scan logical devices ranging from 0 through 255. Red Hat Linux distributions and SuSE SLES 7 support a maximum of 128 SCSI devices as opposed to the SuSE SLES 8 distribution, which supports all 256 SCSI devices. The total of 256 SCSI devices includes any non-EMC storage array devices, such as local disks.

Red Hat Distributions

Red Hat Linux distributions can scan LUN numbers from 0 through 255; however, only the first 128 SCSI devices discovered by the system will be mapped to an `sd` device. Any gaps in devices presented to the system will not result in gaps in the `sd` device names.

The first device can start at a non-zero LUN:

Examples:

- If you have allocated devices 0 through 255 to the Linux host, it will be able to scan up to 255; however, only the first 128 devices (0 through 127) will be usable and accessible by the host.
- If you have LUNs 128 through 255 assigned to a Linux host, you will be able to scan through and use all of them.
- If you have LUNs over 256 exported to a Linux host, the Linux hosts will neither see nor be able to access those devices with an address over 256.

In total, a Red Hat host could support disk devices from `/dev/sda` through `/dev/sddx`.

SuSE Distributions

For SuSE distributions prior to SLES 8, the limitations are the same as for Red Hat distributions. SuSE SLES 8 is different in that it supports up to 256 devices by default as it reserves 16 major numbers for SCSI devices.

Each major number is allocated 0 to 255 minor numbers that are used to denote the partitions for a SCSI device. Therefore, a product of 16 major numbers and 16 minor numbers yields up to 256 SCSI devices. As a result, the kernel is able to scan logical devices ranging from 0 through 255. These disk devices would be reported as `/dev/sda` through `/dev/sdiu`.

Therefore, if you have allocated LUNs from 0 through 255 to the SLES 8 Linux host, the host can scan and use all 256 logical units. However, just as with Red Hat distributions, if you present logical units over 256 to a Linux host, the host will neither see nor be able to access those devices with an address of over 256.

PowerPath Examples

If you have EMC PowerPath® installed, keep in mind that there is a limitation of 128 SCSI devices total per system, *not* per adapter.

- Example without PowerPath — If you have two HBAs or a dual channel HBA and you want both ports/HBAs to see and access devices, you must split the 128 devices between the two HBAs. For example, 64 devices would be allocated to each HBA. If you allocate 128 devices to the first HBA, the second HBA will not be able to access any devices. The limitation of 128 SCSI devices is in the Linux SCSI subsystem and is per system, *not* per adapter.
- Example with PowerPath — Keep in mind that if you present 128 LUNs to a Linux host to one HBA, then only that one HBA will be able to see and access devices.

  In the case of PowerPath, if you have eight HBAs in a server and plan to have two paths per HBA, you will be left with only eight devices per HBA:

  128 LUNs/ (8 HBAs * 2 paths) = 8 devices

Character Device File Names

The corresponding character device file names take the following form:

```
/dev/sgn
```

where `n` begins with zero and increments by one.

The use of the alphabetic sg device file names

```
/dev/sg[a-d][a-z]
```

are now deprecated and are used as links to the sg numeric device filenames.

Current versions of the SymAPI and, therefore, `inquiry` require the existence of the deprecated alphabetic SCSI generic device file names. Not all Linux distributions provide corresponding sg device file names for the sd device file names. Additionally, because the alphabetic sg device file names are deprecated, some distributions, such as SuSE SLES 8 do not provide them at all. For example, on Red Hat releases there are generally 128 sd device file names, 16 sg numeric file names, and 16 sg alphabetic file names. On SuSE SLES 8, there are 256 sd device file names, 64 sg numeric file names, and no sg alphabetic device file names.

In order for `inquiry` to complete, the sg alphabetic file names must exist and be linked to the sg numeric. You can use `mknod` to create any sg numeric file names needed and use `ln` to create the link from the sg numeric to the SG alphabetic.

The Linux kernel assigns minor SCSI device numbers dynamically by assigning them only to devices that are actually connected to the host in the order of their SCSI IDs. This means that connecting an external SCSI device can change the minor numbers of all the internal SCSI devices with a higher SCSI ID.

Dynamic Device Reconfiguration: Procedures

Linux currently lacks an actual command built into the kernel that allows for a dynamic SCSI channel reconfiguration like `drvconfig` or `ioscan`.

The three methods of dynamically reconfiguring a Linux host are:

- Rebooting the host
- Unloading and reloading the modular QLogic driver

- Echoing the /proc filesystem

Rebooting the Host

Rebooting the host allows for reliable detection of newly added devices. The host may be rebooted after all I/O has stopped, whether the driver is modular or statically linked.

Unloading and Reloading the Modular QLogic Driver

Depending upon the hardware configuration and on whether the QLogic driver has been loaded into the kernel as a module, the QLogic driver may be unloaded and then reloaded. If the QLogic driver has been loaded as a module, this unloading and reloading will cause a scan of the SCSI bus. The driver should detect the newly added device(s).

If, however, the modular driver is to be removed from and then reinserted into the kernel, the following must be done before removing the driver:

- If the Naviagent is installed and enabled on the host, then the Naviagent service must be stopped.
- If PowerPath is installed and enabled on the host, then the PowerPath service must be stopped.
- All I/O must be stopped.
- All file systems must be unmounted.

The modprobe command is a wrapper or an extension to insmod and rmmod. The command uses and maintains a set of files that describe all the modules that are available for the current kernel in /lib/modules. The modprobe command can be used to load and unload an entire set of modules. (A *set* of modules can be a single module or a stack of dependent modules.) When modprobe is used, it automatically loads all of the modules needed in a module stack based on the information in modules.dep. EMC recommends using modprobe to load and unload the QLogic modular driver.

The loading of the module is accomplished with either modprobe or insmod. These commands are used to install loadable kernel modules into a running kernel. The commands will attempt to link a module into the running kernel by resolving the symbols from the kernel's exported symbol table.

The unloading of the module can be done with either the modprobe (with the -r switch) or rmmod command. These commands are used to unload the loadable modules from the running kernel if they are

not in use and if other modules are not dependent upon them.

Echoing the /proc Filesystem

If the /proc filesystem is enabled in the kernel, echoing the /proc may be used to add and/or remove single SCSI devices. To add a device, you must first identify the host, channel, target ID and LUN numbers for the device to be added to `/proc/scsi/scsi`.

The command to be run follows this format:

```
echo "scsi add-single-device 0 1 2 3" > /proc/scsi/scsi
```

where:

0 is the host ID.
1 is the channel ID.
2 is the target ID.
3 is the LUN.

This command will add the new device to the file `/proc/scsi/scsi`. If one does not already exist, a device file name may need to be created for this newly added device in the `/dev` directory.

To remove a device, use the appropriate host, channel, target ID, and LUN numbers and issue a command similar to the following:

```
echo "scsi remove-single-device 0 1 2 3" > /proc/scsi/scsi
```

where:

0 is the host ID.
1 is the channel ID.
2 is the target ID.
3 is the LUN.

Dynamic Device Reconfiguration: Device Numbering

In the Linux kernel, the SCSI addresses are not used in the device names as they are in other types of UNIX (Sun, SGI, HP-UX, and BSD, for example). Block device filenames take the form `/dev/sd`*`l`*`n`, where *l* is the letter denoting the physical drive and *n* is the number denoting the partition on that physical drive. Device names and major and minor numbers are assigned dynamically at boot time or device loading time in the order of discovery.

Depending upon the hardware configuration, if a device is added

and the system rebooted, the device numbering might change, possibly rendering the host's mount table inaccurate. For the most consistent results and to reduce the possibility of mount table inaccuracies, new devices should be appended (if possible) to the list of already attached devices. For example, if the host contains multiple HBAs, it would be best to append the new device to the last HBA and to the end of the device list attached to that HBA. This would eliminate the need to alter the preexisting entries in the mount table, since the new device could be appended to that as well. If a new device were added to the first out of two HBAs and the system rebooted, the devices would all shift by one number and the mount table entries would also need to be shifted by one device. If there is only one HBA, the new device can more easily be appended to the list of the regularly attached devices and the mount table altered accordingly.

Linux Naviagent Limitation with fdisk on CLARiiON

There is known issue with the Linux Naviagent in which the Naviagent does not close the SD devices before becoming a daemon and, as a result, the devices remain open. This can cause issues when attempting to access devices.

For example, if the Naviagent service is left running and someone runs `fdisk` to create new partitions on CLARiiON storage array devices, any attempt to make a new filesystem with `mkfs` on the newly created partition(s) will fail.

An example of a warning issued when `fdisk` is run is:

```
WARNING: Re-reading the partition table failed with
error 16. Device or resource busy. The kernel still
uses the old table. The new table will be used at next
reboot.
```

If, after receiving the above warning, the user attempts to run `mkfs` on a newly created filesystem, `mkfs` will also fail with an error similar to the following:

```
mke2fs.ext2: No such device or address while trying to
determine filesystem size
```

This problem can be resolved in one of these ways (described in detail following the list):

- Disable the Naviagent Service.
- Issue a naviagent stop Command

- On RHEL, issue a **`service naviagent stop`** command.
- On SLES 8, issue an **`/etc/init.d/naviagent stop`** command.

◆ Remove the Naviagent RPM.

◆ Rescan the SCSI bus.

Disabling the Naviagent Service

After the Naviagent registers the host with the CLARiiON array, the service can be disabled so that it does not start after rebooting. This can be done manually or by using the setup tool:

◆ Manually:

```
[root@l82bi180 root]# chkconfig --list naviagent
naviagent 0:off 1:off 2:off 3:on 4:on 5:on 6:off

[root@l82bi180 root]# chkconfig --level 345 naviagent off

[root@l82bi180 root]# chkconfig --list naviagent
naviagent 0:off 1:off 2:off 3:off 4:off 5:off 6:off

[root@l82bi180 root]# reboot
```

◆ Using the curses-based setup tool on RHEL:

a. At the command line, type **`setup`** and press ENTER.

The following curses-based menu appears:

```
Text Mode Setup Utility 1.8          (c)1999-2001 Red Hat,Inc.
             +--------+ Choose a Tool +---------+
             |                                   |
             |    Authentication configuration   |
             |    Firewall configuration         |
             |    Keyboard configuration         |
             |    Mouse configuration            |
             |    Network configuration          |
             |    Printer configuration          |
             |    System services                |
             |    Sound card configuration       |
             |    Timezone configuration         |
             |    X configuration                |
             |                                   |
             |       +----------+ +------+       |
             |       | Run Tool | | Quit |       |
             |       +----------+ +------+       |
             |                                   |
             +-----------------------------------+

<Tab>/<Alt-Tab> between elements | Use <Enter> to edit a selection
```

b. Select **Systems services** and scroll down to the naviagent.

c. Use the space bar to deselect the Naviagent service.

d. Quit out of the tool and reboot your host.

Issuing a naviagent stop Command

On RHEL, you can stop the Naviagent service with either of the two following commands and then rerun the **`fdisk/mkfs`** commands:

- **`service naviagent stop`**
- **`/etc/rc.d/init.d/naviagent stop`**

On SLES 8, you can stop the Naviagent service with the following command and then rerun the **`fdisk/mkfs`** commands:

- **`/etc/init.d/naviagent stop`**

Removing the Naviagent RPM

You can remove the Naviagent RPM from the system entirely after it is used for the initial setup of the host on the CLARiiON array.

Example:

```
rpm -ev naviagent-6.6.0.3.8-1
```

Rescanning the SCSI Bus

You can rescan the SCSI bus with a system reboot, or with a removal and reinsertion of the modular driver. (Please refer to *Dynamic Device Reconfiguration: Procedures* on page 2-40.) In either case, all I/O must be stopped and all other mounted file systems must be unmounted before rebooting or removing the modular driver.

If the SCSI bus is not rescanned or the Naviagent service not stopped, attempts to make a new filesystem with `mkfs` on the newly created partition will fail.

Linux Naviagent Limitation with VERITAS Volume Manager on CLARiiON

The Linux Naviagent holds open the devices to be used by VERITAS Volume Manager during the initialization and installation of the CLARiiON devices. When this occurs, the message `Devices busy or in use` is reported.

The issue with the Navaigent and VERITAS Volume Manager can occur on initialization and installation of the VxVM, as well as on subsequent reboots.

Here is an example of the error reported during **`vxinstall`**:

```
  The Volume Manager is now reconfiguring (partition phase)...
  Volume Manager: Partitioning sdq as a new disk.
Device /dev/sdq : does not exist or is in use
ERROR: partition modification failed
Device /dev/sdq : does not exist or is in use
ERROR: partition modification failed
  The Volume Manager is now reconfiguring (initialization phase)...
  Volume Manager: Adding disk01 (sdq) as a new disk.
vxvm:vxdisk: ERROR: Device sdq: online failed:
       Disk is not usable
  Definition of sdq failed. /etc/rc.d/init.d/vxvm-reconfig rootdg sdq
disk01
  The following disks failed initialization:
sdq
```

If the devices were successfully partitioned during the installation, you might also run into an issue after subsequent reboots where the VxVM's `rootdg` becomes unavailable. You will receive errors when trying to issue VxVM commands; this is a result of the configuration daemon (`vxconfigd`) not running, and can happen if the Navisphere host agent is running.

There are two ways to resolve this problem:

- Remove the Naviagent RPM from the system entirely after it is used for the initial setup of the host on the CLARiiON array.

  Example:

  ```
  rpm -ev naviagent-6.6.0.3.8-1
  ```

- Install the Naviagent, but stop the service from starting at boot time. The service can be stopped by using the setup tool as described under *Disabling the Naviagent Service* on page 2-44.

Linux Naviagent Requirement for 64-bit Intel Systems on CLARiiON

The Naviagent for Linux is a 32-bit executable. Therefore, the x86-compat-libs RPM must be installed on 64-bit Intel systems in order for the Linux Naviagent to install correctly.

Compaq Server-Specific Note

When using Compaq systems, it is highly recommended that the Compaq SmartStart CD be run to configure the Compaq server prior to installing the Linux operating system. The SmartStart CD is shipped by Compaq with their systems, and is a bootable CD that is used to configure Compaq servers. If another operating system is selected other than Linux, there may be problems installing the operating system or using the drivers installed in the kernel.

Follow these steps to select Linux as the primary operating system from the Compaq SmartStart CD:

1. Boot the Compaq server using the SmartStart CD.
2. Select the **System Configuration Utility**.
3. After the System Configuration Utility has started, a window titled **Steps in Configuring Your Computer** appears, with the following options listed:

   Step 1: Important System Configuration Information
   Step 2: Add or Remove Boards
   Step 3: View or Edit Details
   Step 4: Examine Switches or Print Report
   Step 5: Save and Exit

   Select **Step 3: View or Edit Details**.

4. In the new window, select **Primary Operating System** (the first item in the list); then select **Unix** from the list of possibilities.
5. Another list of UNIXes appears; select **Linux** from this list.
6. Select **Enter**; then select **<F10> Done**.
7. The console returns to the **Steps in Configuring Your Computer** window. Select **Step 5: Save and Exit**.

   When prompted, select **Save the configuration and restart the computer**.
8. When prompted again to save your configuration and reboot, press ENTER to reboot the system.
9. You may then proceed with installing your operating system.

(CLARiiON Only) Disconnected Ghost LUNs

When a Linux host is attached to both SPs in a CLARiiON storage array, the driver will report a disconnected LUN 0 on SPB and a failure to read the capacity of the device. The Navisphere Host Agent requires that disconnected LUN 0 be reported properly.

For both RHEL and SLES, the s**csi_allow_ghost_devices** parameter must be enabled in `/etc/modules.conf`.

The use of the **scsi_allow_ghost_devices** parameter allows the Linux host to correctly identify the disconnected LUN 0 so that the Navisphere Linux Host Agent can utilize it.

A device file name is allocated to the disconnected LUN 0 in the `/dev` filesystem, but the device cannot be mounted, partitioned, or otherwise accessed.

Here is an example of a disconnected LUN 0 being reported:

```
Jul 17 14:35:57 l82bi134 kernel: sdb : READ CAPACITY failed.
Jul 17 14:35:57 l82bi134 kernel: sdb : status = 1, message = 00, host = 0,
   drivr = 28
Jul 17 14:35:57 l82bi134 kernel: Current sd00:00: sense key Illegal Request
Jul 17 14:35:57 l82bi134 kernel: Additional sense indicates Logical unit not re
dy,cause not reportable
Jul 17 14:35:57 l82bi134 kernel: sdb : block size assumed to be 512 bytes, disk
size 1GB.
Jul 17 14:35:57 l82bi134 kernel: sdb: I/O error: dev 08:10, sector 0
Jul 17 14:35:58 l82bi134 kernel: I/O error: dev 08:10, sector 2
Jul 17 14:35:58 l82bi134 kernel: I/O error: dev 08:10, sector 4
Jul 17 14:35:58 l82bi134 kernel: I/O error: dev 08:10, sector 6
Jul 17 14:35:58 l82bi134 kernel: I/O error: dev 08:10, sector 0
Jul 17 14:35:58 l82bi134 kernel: I/O error: dev 08:10, sector 2
Jul 17 14:35:58 l82bi134 kernel: I/O error: dev 08:10, sector 4
Jul 17 14:35:58 l82bi134 kernel: I/O error: dev 08:10, sector 6
Jul 17 14:35:58 l82bi134 kernel: unable to read partition table
```

(CLARiiON Only) Passive-Not-Ready/ FailOverMode SP Bit Setting and Spinning Up Devices

Two bit settings are required for PowerPath use on CLARiiON arrays. The Array CommPath and FailoverMode bits are enabled by default.

The FailoverMode bit setting provides for Passive-Not-Ready LUN visibility.

The ArrayCommPath bit setting provides a pseudo-logical LUN 0 (LUNZ) for when a LUN 0 is not configured for a host.

If the FailOverMode is enabled on an SP for a Linux host, devices will be reported as *Not Ready*. Depending on the OS, the host may still try to spin up the devices.

The standard Linux behavior is to attempt to spin up the Not Ready devices. This is acceptable Linux SCSI behavior, but a 100-second delay is initiated for each Not Ready device encountered on particular distributions.

Here is an example of this behavior:

```
Jul 17 12:30:06 l82bi134 kernel: sdb: Spinning up
   disk.....................................................................
   ..............................not responding...
Jul 17 12:30:06 l82bi134 kernel: sdb : READ CAPACITY failed.
Jul 17 12:30:06 l82bi134 kernel: sdb : status = 1, message = 00, host = 0,
   driver = 28
Jul 17 12:30:06 l82bi134 kernel: Current sd00:00: sense key Illegal Request
Jul 17 12:30:06 l82bi134 kernel: Additional sense indicates Logical unit not
   supported
Jul 17 12:30:06 l82bi134 kernel: sdb : block size assumed to be 512 bytes,
   disk size 1GB.
Jul 17 12:30:06 l82bi134 kernel: sdb: I/O error: dev 08:10, sector 0
Jul 17 12:30:06 l82bi134 kernel: I/O error: dev 08:10, sector 2
Jul 17 12:30:06 l82bi134 kernel: I/O error: dev 08:10, sector 4
Jul 17 12:30:06 l82bi134 kernel: I/O error: dev 08:10, sector 6
Jul 17 12:30:06 l82bi134 kernel: I/O error: dev 08:10, sector 0
Jul 17 12:30:06 l82bi134 kernel: I/O error: dev 08:10, sector 2
Jul 17 12:30:06 l82bi134 kernel: I/O error: dev 08:10, sector 4
Jul 17 12:30:06 l82bi134 kernel: I/O error: dev 08:10, sector 6
Jul 17 12:30:06 l82bi134 kernel: unable to read partition table
Jul 17 12:30:06 l82bi134 kernel: sdc: Spinning up
   disk.....................................................................
   ..............................not responding...
Jul 17 12:30:06 l82bi134 kernel: sdc : READ CAPACITY failed.
Jul 17 12:30:06 l82bi134 kernel: sdc : status = 1, message = 00, host = 0,
   driver = 28
Jul 17 12:30:06 l82bi134 kernel: Current sd00:00: sense key Illegal Request
Jul 17 12:30:06 l82bi134 kernel: Additional sense indicates Logical unit not
   supported
```

To avoid this behavior and the delay that is incurred by Linux trying to spin up the device, it is necessary to disable both the FailOverMode and ArrayCommPath bit settings if PowerPath is not being used.

If PowerPath is being used, ensure that both the FailOverMode and ArrayCommPath bit settings are enabled. These settings are required for PowerPath use.

Refer to the Navisphere Management software documentation for instructions on how to disable these bit settings.

Configuring and Compiling the Kernel

EMC highly recommends that the kernel not be compiled. There are other modular methods to enable parameters as mentioned previously in this document.

By applying a patch and compiling the kernel, the support contract with the Linux distributor will be violated.

If necessary or otherwise required for a new installation, proceed to *New Installation* later on this page.

Install Linux from the distribution installation CD (including sources), following the procedure provided in the installation guide. Partition the boot drive and select the packages and services necessary for the host.

New Installation

Follow these steps to configure the host in a new installation:

1. Prior to configuring the kernel, change to the appropriate directory:
   - If using Red Hat:

     ```
     cd /usr/src/linux-2.4
     ```

   - If using SuSE:

     ```
     cd /usr/src/linux
     ```

2. Configure the kernel, using one of these commands
   - **`make config`**
   - **`make menuconfig`**
   - If using the GUI: **`make xconfig`**
3. From the Main Menu, make the changes necessary to your configuration and save them.
4. Compile the kernel:
   - If using Red Hat distributions:

     ```
     make dep clean install modules modules_install
     depmod -ae
     ```

- If using SuSE distributions:

```
make dep clean bzImage modules modules_install
cp System.map /boot/System.map-2.4.19
cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.19
depmod -ae
```

5. If desired, build a ramdisk image:

```
cd /boot
```

Depending on the distribution and kernel versions, use one of the following (where *$1* is the kernel version):

- For Red Hat distributions use:

```
mkinitrd /boot/initrd-$1.img $1
```

Example:

```
mkinitrd /boot/initrd-2.4.16.img 2.4.16
```

- For SuSE distributions use:

```
mk_initrd -k vmlinuz-$1 -i initrd-$1
```

Example:

```
mk_initrd -k vmlinuz-2.4.16 -i initrd-2.4.16
```

The **mkinitrd/mk_initrd** command is used to create a new ramdisk image and is dependent upon the loopback device being available. If the loopback driver is selected in the kernel as a module, ensure that it is loaded with **lsmod**. If the loopback driver was not loaded, insert it with **insmod loop**.

6. It might be necessary to add the new kernel and ramdisk images to the appropriate bootloader configuration file prior to rebooting:

a. If LILO is being used, add the new kernel version to `lilo.conf` so it will be bootable:

```
vi /etc/lilo.conf
```

Example:

```
image=/boot/vmlinuz-2.4.16
       label=linux-2.4.16
       root=/dev/sda1
       initrd=/boot/initrd-2.4.16.img
       read=only
```

LILO must be run prior to rebooting in order to update the loading map:

`lilo -v`

b. If GRUB is being used, add the new kernel version to `grub.conf` so it will be bootable.

For Red Hat distributions, use:

`vi /boot/grub/grub.conf`

Example:

```
title Red Hat Linux (2.4.9-34custom)
         root (hd0,0)
         kernel /vmlinuz-2.4.9-34custom ro root=/dev/sda5
         initrd /initrd-2.4.9-34custom.img
title Red Hat Linux (2.4.9-34enterprise)
```

For SuSE distributions, use:

`vi /boot/grub/menu.lst`

Example:

```
title linux
         kernel (hd0,0) /vmlinuz root=/dev/sda2 vga=788
         initrd (hd0,0) /initrd-mod
```

GRUB does not have to be rerun after having made changes to this file.

7. Reboot the host.

Incorporating EMC Fibre Channel

Once the EMC storage array has devices assigned with device addresses and the directors have been switched on line, the host can see all the target EMC devices assigned to that host interface. Devices are presented to the host in the same manner as devices accessed through a standard SCSI interface.

3

Configuring a Boot Device on the EMC Storage Array

EMC supports booting Red Hat Linux from an EMC storage array through an EMC-qualified QLogic Fibre Channel HBA. (Refer to the *EMC Support Matrix* for specific HBAs, BIOS revisions, and drivers.)

- Cautions and Restrictions 3-2
- Preparing the CLARiiON Storage System 3-5
- Configuring the Boot BIOS 3-6
- Installing Red Hat Enterprise Linux 2.1 onto the Boot Device with the QLogic v6.x Series Driver 3-8

Cautions and Restrictions

CAUTION

If Linux loses connectivity long enough, the disks disappear from the system. To prevent further data from being lost in a situation like this, EMC recommends that the error behavior be changed from *continue* to *remount read-only*. To make this change, consult the manpage for tune2fs. A hard reboot is required to bring the system back to a usable state.

CAUTION

You should configure the host with enough memory to eliminate the need for a swap partition. EMC cannot recommend an amount of memory as this varies greatly depending upon the environment.

EMC recommends shutting down the host server during any maintenance procedures that could make the boot disk unavailable to the host.

Any of these events could crash a system booting from an EMC storage array:

- Lost connection to storage array (pulled or damaged cable connection)
- Storage array service/upgrade procedures, such as on-line Symmetrix microcode or CLARiiON FLARE upgrades and/or configuration changes
- Symmetrix director failures or CLARiiON SP failures, including failed lasers
- Storage array power failure
- Storage Area Network failures, such as failures in Fibre Channel switches, switch components, or switch power
- Storage Area Network service/upgrade procedures, such as firmware upgrades or hardware replacements
- Trespass of a CLARiiON LUN from one SP to the other.

Limitations

Boot configurations must not deviate from the following limitations established by EMC:

- A maximum of 32 volumes can be configured on the EMC storage array port that will be used to boot Linux.
- When attached to a Symmetrix, the physical-to-logical split must be 1:8 or less (1:7, 1:6, etc.). EMC supports the use of any LUN from 00 through FF as a boot disk with QLogic adapters.
- EMC recommends that you install, rather than upgrade, the kernel-source and kernel UP, SMP, and enterprise RPMs.
- Hosts booting from an EMC storage array are limited to the use of one HBA.
- Hosts booting from an EMC storage array must use the QLogic driver as a module, not statically linked into the kernel.
- The Symmetrix or CLARiiON device that is to contain the Master Boot Record (MBR) for the host must have a lower logical unit number (LUN) than any other device visible to the host. This device must be mapped as `/dev/sda` by the Linux operating system for the boot to succeed from the device.

  Note that this includes the following cases:

  - Volume Logix database device when using Volume Logix in a Symmetrix environment — The installer will fail to write the MBR to this device, as it is write-protected. EMC recommends that you map the Volume Logix database device so that it is the highest LUN presented to the host, to avoid problems arising from a write-protected device being presented to the host.

    Please be aware that the LUN number should not be higher than 254 (FE) if it is to be used by ESN Manager running on a Microsoft host.
  - CLARiiON ghost LUN — If the LUN 0 presented to a SCSI path is not owned by the SP attached to that path, a disconnected LUN 0 (ghost LUN) will be presented. The ghost LUN is not write-enabled so that the MBR cannot be written to it. Always ensure that the boot LUN is owned by the correct SP and does not trespass at any time; trespass of the boot LUN would result in a system crash.

- The installer installs the boot block onto only the first disk it sees. This includes SCSI and IDE drives in your system. If there are only EMC storage array devices attached to the system, the boot BIOS should be set to the first LUN the HBA sees. If there are other local disks in the system, you may be forced (depending on how Linux discovers the drives) to put your boot block onto a local disk.

Preparing the CLARiiON Storage System

Boot device support requires a Navisphere Management station with Navisphere Manager or CLI. The station must be separate from the boot server, but networked to the storage system (FC4700 and later model arrays).

Before you can install Linux onto a storage system, the storage system must have at least one bound LUN in a Storage Group for the boot server. The LUN must be owned by the SP connected to the boot HBA.

The Linux host system boot LUN must be assigned Host LUN ID 0.

During this installation procedure, you should have only one LUN in the boot Storage Group, so that you can easily identify the boot LUN. Later, you can add other LUNs to this Storage Group.

If you need to create a LUN, refer to the EMC ControlCenter Navisphere Management software documentation for your array type.

Configuring the Boot BIOS

After the boot BIOS is installed and enabled, you must configure it for your boot device. The following procedure describes how to configure the boot BIOS. The procedure assumes that the boot BIOS is already installed onto the adapter(s).

1. Connect the EMC storage array boot port to the adapter in the lowest-numbered PCI slot in the server. For example, if you have three adapters in the system in slots 2, 4, and 5, connect the cable to the adapter in slot 2. Do not connect cables to the other adapters at this time.
2. Boot the server, and press CTRL-Q immediately when you see a banner similar to the following:

```
Qlogic Corporation
QLA2312 PCI Fibre Channel ROM BIOS Version 1.42
Copyright © Qlogic Corporation 1993-2002 All rights reserved
Press <CTRL - Q> for Fast!UTIL
www.qlogic.com
```

   The banner display corresponds to the BIOS revision pertinent to the HBAs currently installed. Refer to the *EMC Support Matrix* for specific HBAs and BIOS revisions.

3. After `Fast!UTIL` loads, the display depends on whether there are multiple QLogic HBAs installed:
   - If there is only one QLogic HBA, the `Fast!UTIL` **Options** menu appears.
   - If there are multiple QLogic HBAs, a list of addresses occupied by those HBAs appears. Since the EMC storage array is attached to the lowest-numbered PCI slot, select the first adapter from the list; then press ENTER. The `Fast!UTIL` **Options** menu appears.
4. From the `Fast!UTIL` **Options** menu, select **Configuration Settings** and press ENTER.
5. From the **Configuration Settings** menu, select **Adapter Settings** and press ENTER.
6. From the **Adapter Settings** menu, select **Adapter BIOS** and press ENTER to enable it if it is not already enabled.
7. Press ESC to exit the **Configuration Settings** menu.

8. From the **Configuration Settings** menu, select **Selectable Boot Settings** and press ENTER.
9. From the **Selectable Boot Settings** menu, select **Selectable Boot Device** and press ENTER to enable it if it is not already enabled.
10. Select **Current Boot Node Name** and press ENTER.

    The adapter scans for attached storage devices, and displays them on the screen.
11. Choose the storage array port from which you wish to boot. Its entry will be similar to the following:

```
ID  VENDOR     PRODUCT       REV       NODE NAME          Port ID
12  EMC        Symmetrix     5568      50060482BFD06C02   0000E4
```

    Select your boot device and press ENTER.
12. A list of LUNs appears in a new window. Select the LUN from which you wish to boot and press ENTER.
13. Press ESC at the **Selectable Boot Settings** menu to return to the **Configuration Settings** menu.
14. Press ESC at the **Configuration Settings** menu to return to the **Options** menu.

    Select **Save Changes** and press ENTER.
15. Press ESC to exit Fast!UTIL.
16. Reboot the host.

Installing Red Hat Enterprise Linux 2.1 onto the Boot Device with the QLogic v6.x Series Driver

The Red Hat Enterprise Linux (RHEL) 2.1 Advanced Server (v2.4.9-e.3) and Enterprise Server (v2.4.9-e.12) distributions ship with version v5.31 of the QLogic driver. This version of the driver does not support fabric boot; therefore, you must use the driver disk installation method to use a LUN on the storage array as the boot device.

This section describes the steps that must be taken to successfully perform the installation environment for both Symmetrix and CLARiiON arrays. Please be sure to perform the preparatory steps earlier in this chapter. Also, refer to *Limitations* on page 3-3 to familiarize yourself with the restrictions inherent to booting from the array in a Linux environment.

The installation process consists of the following distinct components:

1. Building the driver disk — The driver disk must be built for the correct hardware and software combination.
2. Installing Red Hat Enterprise Linux 2.1— The installation makes use of the driver disk to discover the devices that have been set up previously.
3. Upgrading to the appropriate errata — The final step is to update the host to the required errata level.

After these steps are completed, the host system is ready for use.

Building a Driver Disk

The driver disk provides the correct driver and identifying information needed for the Red Hat Installer to access the devices presented over the fabric. The driver disk installation method allows for devices to be discovered at installation time, for which the device drivers are not part of the standard distribution. Also, this method can be used to provide an updated version of a driver that is part of the standard distribution, such as is the case here.

The process of building a driver disk for Red Hat Linux is described in detail on several websites. In this section, the key steps and specific requirements for building the particular QLogic HBA driver disks

will be given. The following websites provide additional details and supplementary information on this process:

- `http://people.redhat.com/dledford` — Doug Ledford's website contains the kit needed to build driver disks, as well as detailed information on the function and development of driver disks.
- `http://www.qlogic.com/support/drivers_software.asp` — QLogic's support website provides information on how to build a driver disk with an example for a QLogic driver. This will be useful in making the correct edits.

The entire build process must be performed on a Red Hat server, which contains the kernel source tree for the target version of Red Hat Linux, for which the driver disk is being built. In this case, this means that the v2.4.9-e.3 or v2.4.9-e.12 kernel source tree needs to exist. The most straightforward method for ensuring that the necessary source tree exists is by installing kernel-source from the distribution Red Hat Enterprise Linux 2.1 distribution that you will be installing onto the storage array.

Note that it is possible to have multiple source trees on a driver disk build server to allow for building disks for multiple targets; this feature is not needed for this driver disk build.

Procedure

The following are the steps to build driver disks for the Red Hat Enterprise Linux 2.1 installation:

1. Create a base directory (for example, `/opt/dd`), under which you will be performing the driver disk work. Make sure that at least 30MB of free space is available in the filesystem that contains this directory. Change (**`cd`**) to the directory you just created.
2. Download `mod_devel_kit` from `http://people.redhat.com/dledford`. On this web page you will see a link to **Download the current Device Driver Update Disk Devel Kit**. Click on this link to download `mod_devel_kit` into the directory that you created in step 1.
3. Untar the file `mod_devel_kit.tgz`. This will create a `mod_devel_kit` directory tree automatically.
4. Read through the `README` and `README.new_kernel_versions` files to familiarize yourself with the process of creating driver disks.

5. Create a separate directory (for example, `/opt/dd/example`), under which you will store the download and untar QLogic's example of building a driver disk for the qla2300 driver. Make sure that at least 62MB of free space is available in the filesystem that contains this directory. Change (**`cd`**) to this newly created directory.

6. Use a web browser to download the example kit from the following URL:

```
http://download.qlogic.com/drivers/10572/how_to_build_dd_kit.tgz
```

   This example kit is the same for both the QLA2200 and QLA2300 series HBAs.

7. Untar `how_to_build_dd_kit.tgz`, which will create three files in the directory.

   Untar `mod_devel_kit_RH8.0.tgz`, which will create a fully populated `mod_devel_kit` directory tree like the one created in step 3. Note that this is an example tree set up for Red Hat 8.0. We will solely use this tree for several example files.

8. Change (**`cd`**) back to the first driver disk tree (for example, `/opt/dd`) and then go into the `mod_devel_kit` directory. From this point forward use this as your current directory, unless otherwise specified.

9. Import the kernel source tree for which the driver disk kit will be built. Use the following command:

```
# make IMPORT_TREE=/usr/src/linux-2.4.9-e.3 IMPORT_VER=2.4.9-e.3 import
```

   This command copies the appropriate files from the `/usr/src/linux-2.4.9-e.3` directory tree to `2.4.9-e.3` under the `mod_devel_kit` directory. The example shows the values for the RHEL 2.1 Advanced Server distribution; for RHEL 2.1 Enterprise Server use e.12 instead of e.3.

10. In the current directory (`mod_devel_kit`) edit Makefile and make the following modifications:

    a. Uncomment the variable definition that reads `DRIVERS=scsi` and make sure that all other `DRIVERS` variable definitions are commented out.

b. Uncomment and edit the line that defines `AS21_KERNEL_DIRS`, so that it reads `AS21_KERNEL_DIRS=2.4.9-e.3` or `2.4.9-e.12`, as appropriate, and make sure that all other `KERNEL_DIRS` definitions are commented out.

11. From the `mod_devel_kit` directory create the `scsi` directory.

    `mkdir scsi`

12. Copy the EMC-approved driver for your HBA (for example, `qla2x00src-v6.04.02.tgz`) into the `scsi` directory. Refer to *Downloading the Driver* on page 2-17 for the procedure to obtain the driver.

13. Set your current directory to `scsi`, and untar the driver kit.

    `cd scsi`
    `tar zxvf qla2x00src-v6.04.02.tgz`
    (substitute the correct file name)

14. Now it is time to change some of the files to ensure that the correct driver is built during the generation of the driver disks.

    First, delete the files `Makefile.kernel` and `makefile` from the directory to eliminate potential conflicts.

    The particular files that need to be updated are:

    a. `Makefile` — Copy the file from the sample directory tree to the scsi directory:

       `cp /opt/dd/example/mod_devel_kit/scsi/Makefile .`

       Edit the file to update the driver version parameter, **DRV_VER**. Set this to the correct value; grep `VERSION qla_version.h` to check the version number.

    b. `modinfo` — Copy the file from the sample directory tree to the scsi directory:

       `cp /opt/dd/example/mod_devel_kit/scsi/modinfo .`

       Edit `modinfo` to update the driver version. Also set the kernel version to v2.4.9-e.3 or v2.4.9-e.12. For example:

       `Updated qla2300 driver (v6.04.02) for AS21-2.4.9-e.3`

    c. `disk-info` — Copy the file from the sample directory tree to the `scsi` directory:

       `cp /opt/dd/example/mod_devel_kit/scsi/disk-info .`

No edits are needed unless you are building a QLA2200 driver disk.

d. `pcitable` — Copy the file from the sample directory tree to the `scsi` directory:

```
cp /opt/dd/example/mod_devel_kit/scsi/pcitable .
```

Edit this file only if a QLA2200 driver disk is being built. In that case, change the product ID on the first line (0x2300) to 0x2200 and edit the remainder of the line to reflect the QLA2200 rather than the QLA2300. Also delete the second line.

e. `modules.dep` — Copy the file from the sample directory tree to the `scsi` directory:

```
cp /opt/dd/example/mod_devel_kit/scsi/modules.dep .
```

No edits are needed.

15. Copy the following files from the appropriate kernel source tree into the `scsi` directory:

```
scsi.h
scsi_module.c
hosts.h
sd.h
scsi_obsolete.c
scsi_obsolete.h
```

For example with your current directory still `scsi`:

```
cp /usr/src/linux-2.4.9-e.3/drivers/scsi/sd.h .
```

16. Set your current directory back to `mod_devel_kit` and run make:

```
make
```

This builds the `modules.cpio-gz-xxxx` files for the various architectures and put these files into the `as21` directory.

17. Create the driver disks by issuing this command:

```
make disk
```

This command creates the `dd.img-xxxx.gz` files for each of the architectures in the `as21` directory.

18. Select the correct file for your architecture, decompress it and copy it to a floppy disk. For example:

```
gzip -d dd.img-i686.gz
dd if=dd.img-i686 of=/dev/fd0 bs=1440k
```

You have now successfully created the driver disk required for your installation. Insert this disk into the server when prompted for it during the installation.

Installing Red Hat Enterprise Linux 2.1 onto the Storage Array

Use the driver disk that was created using the procedure in the previous section for the installation of Red Hat Enterprise Linux 2.1 onto a device presented by your EMC storage array.

Before starting the installation, you should have performed the steps described under *Configuring the Boot BIOS* on page 3-6.

To install Red Hat Enterprise Linux 2.1 onto the storage array:

1. Boot the server to be installed from either CD-ROM disk 1 of the Red Hat Enterprise Linux 2.1 Advanced Server distribution or a boot diskette.
2. At the `boot:` prompt, type **`linux dd`**, and press ENTER.
3. You are prompted:

   ```
   Do you have a device driver disk?
   ```

   Select **YES**.
4. You are prompted:

   ```
   Insert your driver disk and press "OK" to continue.
   ```

   Insert the device driver disk, and select **OK**.
5. When asked for the type of installation to perform, select **Custom**.

   This selection allows for complete control of the boot device selection and manipulation.

   From this point, follow your normal installation process.

Although EMC makes no recommendations on any particular installation selections, it is worthwhile creating a boot diskette for this installation, to aid in troubleshooting any connectivity problems.

On completion of the installation, your host will boot into Red Hat Enterprise Linux 2.1.

Upgrading Kernel Errata

At this point, your host will be running Red Hat Enterprise Linux 2.1 Advanced Server v2.4.9-e.3 or Enterprise Server v2.4.9-e.12. You will want to upgrade the kernel errata to take advantage of fixes and features incorporated in the newer errata.

When you have installed onto the array, you should upgrade to a minimum of kernel v2.4.9-e.24. This errata release has the requisite capabilities built into the QLogic driver that facilitate booting from the array. Follow the normal installation process for this errata release. Once installed, your host will boot without the need for any special steps.

Remember to install the kernel packages, rather than upgrading them so that you have the capability to boot either kernel version. Also, if you have created a boot diskette, you will need to create an updated diskette after the upgrade.

Index

A

addressing 1-2
API libraries, SNIA 2-27
AX100 storage arrays 2-17

B

binding, persistent 2-28
BIOS
  boot 3-4
  settings 2-6, 2-7
  upgrading using NVRAM file 2-8
  version 2-6
boot BIOS, configuration 3-6
boot block 3-4
boot configuration 3-3
boot device
  cautions and restrictions 3-2
  CLARiiON preparation 3-5
boot HBA 3-5

C

CLARiiON
  configuration 1-3
  storage arrays 2-17
Compaq SmartStart CD 2-47
configuration
  boot 3-3
  storage array 1-3
  utilities 1-3
connectivity 1-2

D

data rate, QLA23xx, setting manually 2-9
default HBA settings 2-5
device access protocol 1-2
device drivers 2-37
devices
  storage array 1-3
driver disk 3-8
  Red Hat Linux 3-8
drivers
  sd driver 2-37
  sg driver 2-37
  sr driver 2-37
  st driver 2-37
dynamic reconfiguration 2-40

E

EMC NVRAM file 2-7
EMC settings, HBA 2-7

F

fdisk 2-43
Fibre Channel directors 1-2
firmware settings, HBA 2-6
functions, Linux 1-3

H

HBA
  boot 3-5
  default settings 2-5
  EMC settings 2-7
  FC-AL Loop ID 2-4

firmware settings 2-6
installation 2-2
loop ID 2-4

I

information
initiator 2-28
target 2-28
initiator information 2-28
installation, QLogic HBA 2-2

K

kernel, configuring and compiling 2-51

L

Linux hosts, booting 1-2
Linux Naviagent 2-43, 2-47
loop ID, HBA 2-4

M

Master Boot Record (MBR) 3-3

N

Naviagent
Linux 2-43, 2-47
with fdisk 2-43
with VERITAS Volume Manager 2-46
Naviagent service, stopping 2-45
NVRAM file 2-5
EMC 2-7

P

parameter, queue depth 2-30
performance enhancement 2-30
persistent binding 2-28
protocol, device access 1-2

Q

queue depth parameter 2-30

R

reconfiguration, dynamic 2-40

S

settings
EMC NVRAM file 2-7
NVRAM file 2-5
QLogic BIOS 2-7
sg driver 2-37
single initiator zone 1-2
SNIA API libraries 2-27
storage array
configuration 1-3
devices 1-3
storage arrays, CLARiiON 2-17
Symmetrix configuration 1-3
system crash events 3-2

T

target information 2-28
topology, setting 2-11

U

utilities, configuration 1-3

V

VERITAS Volume Manager 2-46
versions, BIOS 2-6

Z

zoning 1-2